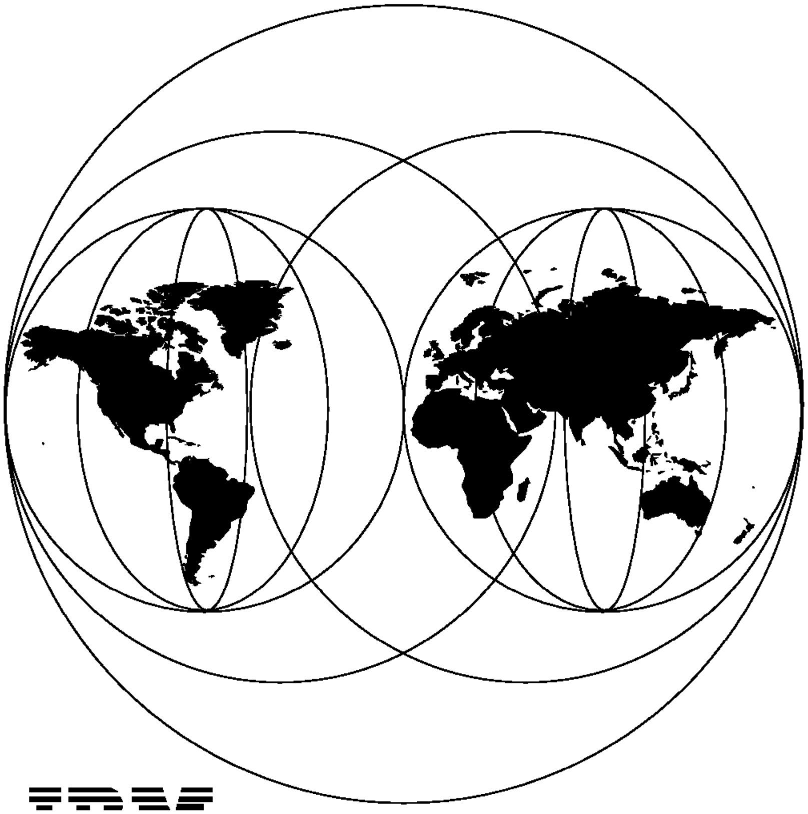
SG24-4750-00

IMS/ESA Multiple Systems Coupling in a Parallel Sysplex

April 1997



International Technical Support Organization
San Jose Center

International Technical Support Organization

SG24-4750-00

IMS/ESA Multiple Systems Coupling in a Parallel Sysplex

April 1997

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix C, “Special Notices” on page 99.

First Edition (April 1997)

This edition applies to Version 5, Release Number 1 of IMS/ESA, Program Number 5695-176, as well as the enhancements introduced by IMS/ESA Version 6, Release Number 1, Program Number 5655-158.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. QXXE Building 80-E2
650 Harry Road
San Jose, California 95120-6099

Contents

Figures

Tables

Preface

This document provides IMS/ESA system programmers and application designers with guidelines on using IMS/ESA Multiple Systems Coupling (MSC) in an IMS/ESA sysplex environment. It focuses on operations, MSC user exits, performance evaluation, and information useful for problem determination. Some knowledge of IMS/ESA is assumed.

This redbook is intended to aid information technology professionals in understanding the issues involved in establishing an IMS/ESA MSC environment, both in the sysplex and non-sysplex environments.

The book outlines the reasons for considering an MSC environment, offers advice on how to go about implementing the environment, and describes what performance to expect.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization San Jose Center.

Gary Wicks is an Advisory Systems Engineer in IBM Canada. He has 23 years of experience in the field of software support and technical support planning and management. He holds a degree in math and physics from the University of Toronto. His areas of expertise include diagnostic and implementation support for IMS/ESA customers. He has written extensively on IMS/ESA diagnostics, IMS MSC in sysplex configurations, and application and database design in data sharing environments.

Jean Pierre De Villiers is an Advisory Technical Specialist in IBM South Africa. He has eight years of experience in the IMS field. He holds a degree in economics from Rand Afrikaans University. His area of expertise is IMS systems programming.

Thanks to the following people for their invaluable contributions to this project:

Attila Fogarasi
Geoff Nicholls
ITSO Specialists for IMS at the Application Development and Data Management ITSO Center, San Jose

Cedric Chen
Pat Smith
Don Terry
Mark Ziebarth
IBM Santa Teresa Laboratory

Bob Gendry
IBM Dallas Systems Center

A special thanks to the following customers for reviewing the draft of this book and providing valuable comments:

Kevin Stewart
Toronto Dominion Bank, Toronto, Ontario, Canada

Jeung-Ho Kim
Paul Higgins
Canadian Imperial Bank of Commerce, Toronto, Ontario Canada

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 111 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Home Pages at the following URLs:

For Internet users	`http://www.redbooks.ibm.com`
For IBM Intranet users	`http://w3.itso.ibm.com/redbooks`

- Send us a note at the following address:

  `redbook@vnet.ibm.com`

Chapter 1. Multiple Systems Coupling: An Introduction

As the size of the workload processed by IMS Transaction Manager and Database Manager (TM and DB) systems increases, the need to distribute and balance the workload across multiple systems becomes more of a requirement. IMS/ESA provides two facilities that facilitate communication between subsystems and hence the distribution of processing:

- Intersystem Communication (ISC)
- Multiple Systems Coupling (MSC)

This book focuses on the MSC option, looking in particular at the performance of MSC as it distributes transaction and message traffic between IMS subsystems.

MSC is a private IMS/ESA protocol that permits the coupling of IMS/ESA systems to other IMS/ESA systems only. MSC is an extension of the IMS/ESA communication and scheduling capabilities and enables the user to perceive one virtual IMS system, when in fact, behind the scenes, transactions are being routed among a complex of IMS subsystems. The design is based on automatic transaction and message routing, without any required application interaction.

Each system is assigned a unique identifier known as the systems identifier (SYSID). During the system generation process, transactions are defined as either local (processed on this system) or remote (processed on the system identified by the SYSID on the transaction or the application definition). To maintain application transparency for change (CHNG) calls to alternate program communication blocks (PCBs), remote terminal definitions called *remote LTERMs* are created by associating logical terminals (LTERMs) with a given link definition.

A feature called *directed routing* enables applications to be modified to understand the MSC environment. The use of directed routing reduces the IMS system generation effort. Regardless of the process used, the responses from the executed transaction can be returned to the terminal that originally entered the transaction or to other terminals. User exits are in wide use among customers who use MSC.

1.1 Reasons for Using MSC

Multiple systems are linked together by MSC under one system image to:

- Provide load sharing by spreading a large terminal network among several IMS/ESA systems.
- Distribute transaction processing when affinities exist and the system receiving the input does not have control of the necessary resources to execute the transaction and access the data.
- Raise the general level of performance by distributing the workload if system processing requirements exceed those available on a single CPU processor.
- Align processing sites to distributed business center organizations.

1.2 MSC-Supported Facilities

MSC supports four types of physical links between IMS/ESA systems (see Figure 1):

BSC	binary synchronous
CTC	channel to channel
MTM	main storage to main storage
SNA	VTAM SDLC link

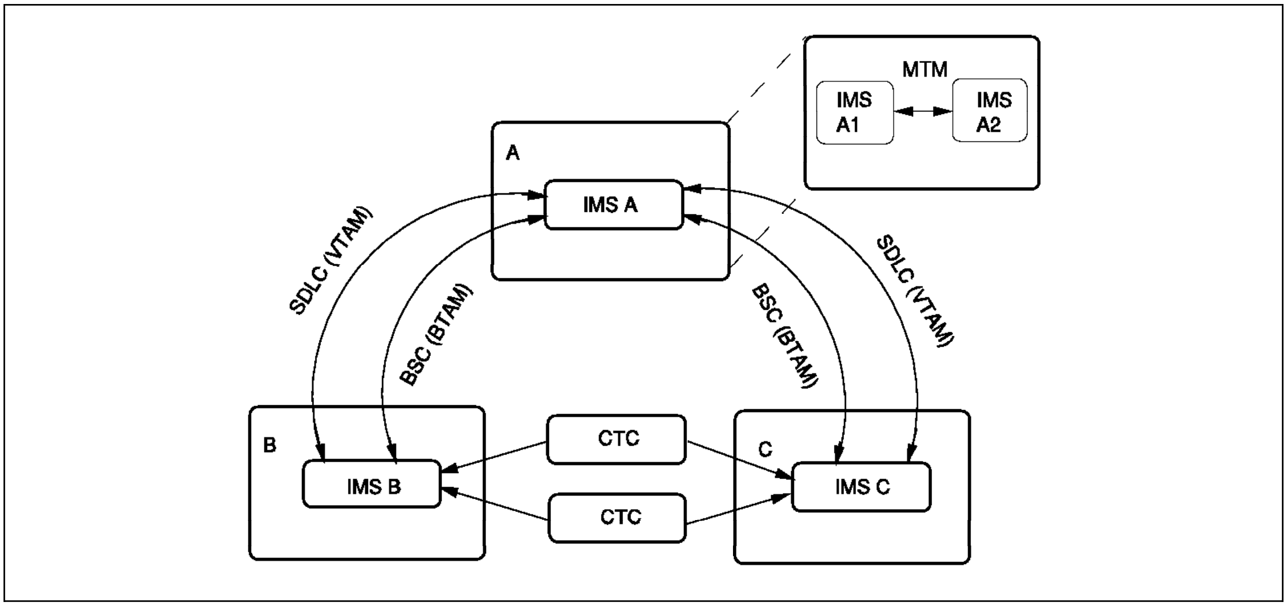


Figure 1. Multiple Physical Links with Several Processors

The method of linkage between IMS/ESA systems may consist of any combination of SNA, MTM, CTC, and BSC links, which have to be defined during the IMS/ESA system definition. Only the BSC line, VTAM SDLC link, and the CTC adapter represent actual hardware links. The MTM link is a software link between IMS/ESA systems controlled by the same operating system and is intended for test purposes.

To connect more than two CPUs, one or more intermediate node CPUs are incorporated into the linkage structure. Intermediate nodes are transparent to any application, and IMS/ESA handles the flow through these nodes.

The full recovery and restart capabilities of IMS/ESA apply equally to MSC traffic.

MSC permits the steps of a conversation to be distributed over multiple IMS/ESA subsystems, transparent to both the source terminal operator and to each conversational step in the application. Program-to-program switches and message switches are also supported.

LU6.2 application programs can process transactions on remote IMS/ESA systems. Extended Terminal Option (ETO) also supports remote LTERMs.

MSC does not support Fast Path. The expedited message handler (EMH) does not have access to the message queue, which is required by MSC operation.

1.3 Overview of MSC Transaction Flow

The flow of a transaction in an MSC environment has additional steps beyond those of the flow of a single-system environment. These steps are illustrated in Figure 2 on page 4:

- A transaction defined as remote on this local system, entered from a terminal (1), is placed on the message queue of the local system (2).
- MSC support removes the message from the queue (3).
- The message is sent across the MSC link (4).
- The message is placed on the message queue of the remote system (5).
- The message waits to be processed by the application program (6).
- When the application program processes the message, it sends a reply to the originating IMS system, as indicated by the SYSID in the message.
- The reply message is placed first on the message queue (7) in the processing system, with a destination of the input LTERM (8).
- MSC support removes the message from the message queue (9).
- MSC sends the message back across the MSC link (10).
- MSC places the message on the message queue of the originating system (11).
- MSC sends the message to the input terminal (12).

The data and control flows are discussed in more detail in Chapter 6, “Identifying and Solving MSC Problems.”

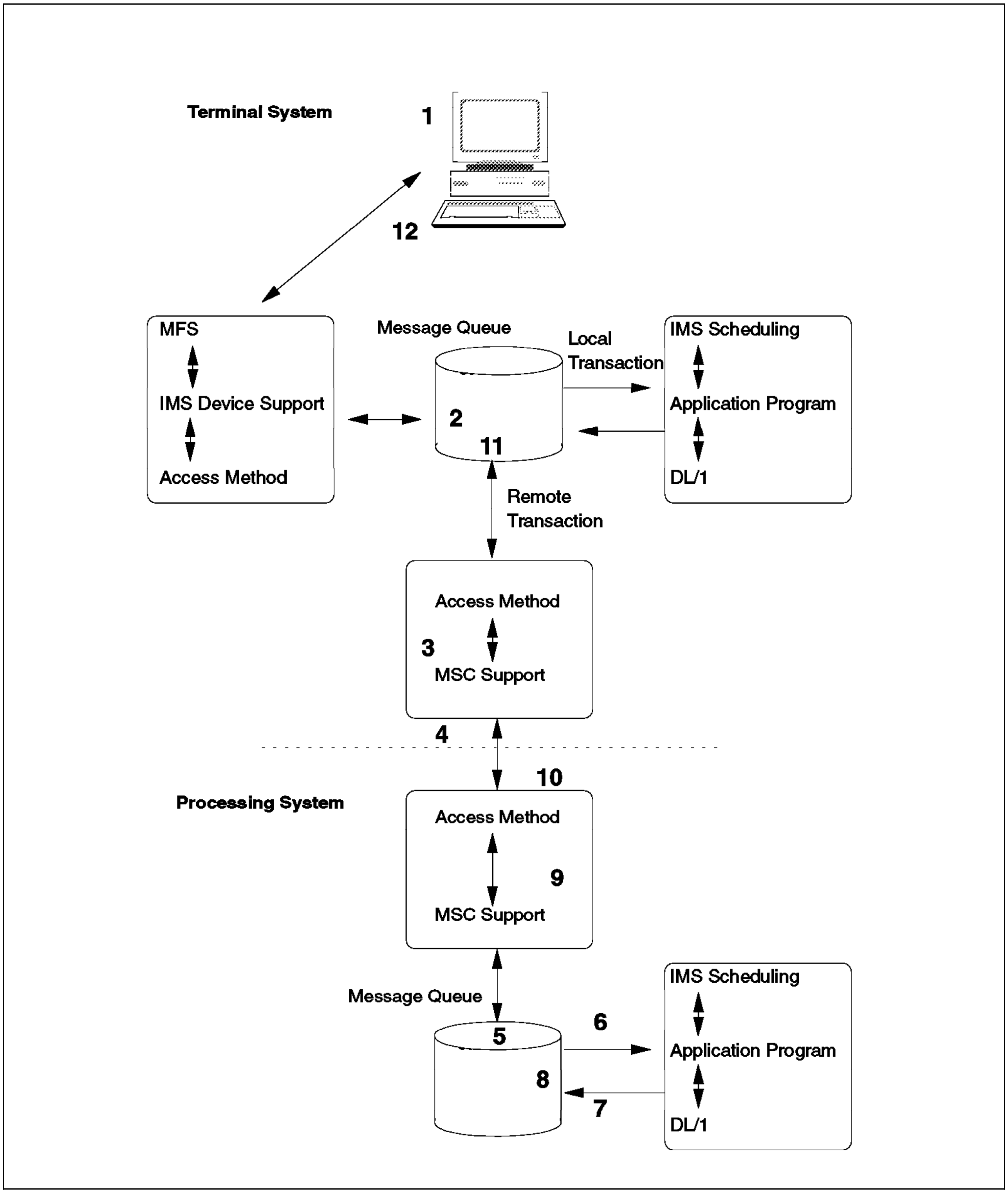


Figure 2. Multiple IMS Systems — Transaction Flow

Chapter 2. MSC in a Sysplex Environment

The concept of a parallel sysplex has been around since the early 1990s, but it became a practical computing facility once MVS and its core products were fully integrated in support of that platform. With IMS/ESA V5.1, customers can begin to exploit the facilities of the sysplex environment.

We examine briefly the parallel sysplex before presenting possible scenarios in which MSC can be used to connect the systems in the sysplex.

2.1 The Parallel Sysplex

A parallel sysplex is a collection of MVS/ESA systems that are combined through a set of coupling and data sharing facilities to provide a single-image computing facility that exploits the latest microprocessor technology and the advantages of cooperative, parallel processing.

2.1.1 Objective of a Parallel Sysplex

The primary objective of a parallel sysplex is simply to provide a price-competitive computing facility for large commercial processing systems. When you consider the parameters within which this objective was reached, you can appreciate the significance of the achievement. The parameters were these:

- Preserve a single system image.
- Provide investment protection (hardware, software and skill).
- Ensure that current applications benefit.
- Provide *significant* room for growth.
- Improve continuous availability.
- Make it easy to use.

Figure 3 on page 6 presents the components of a parallel sysplex.

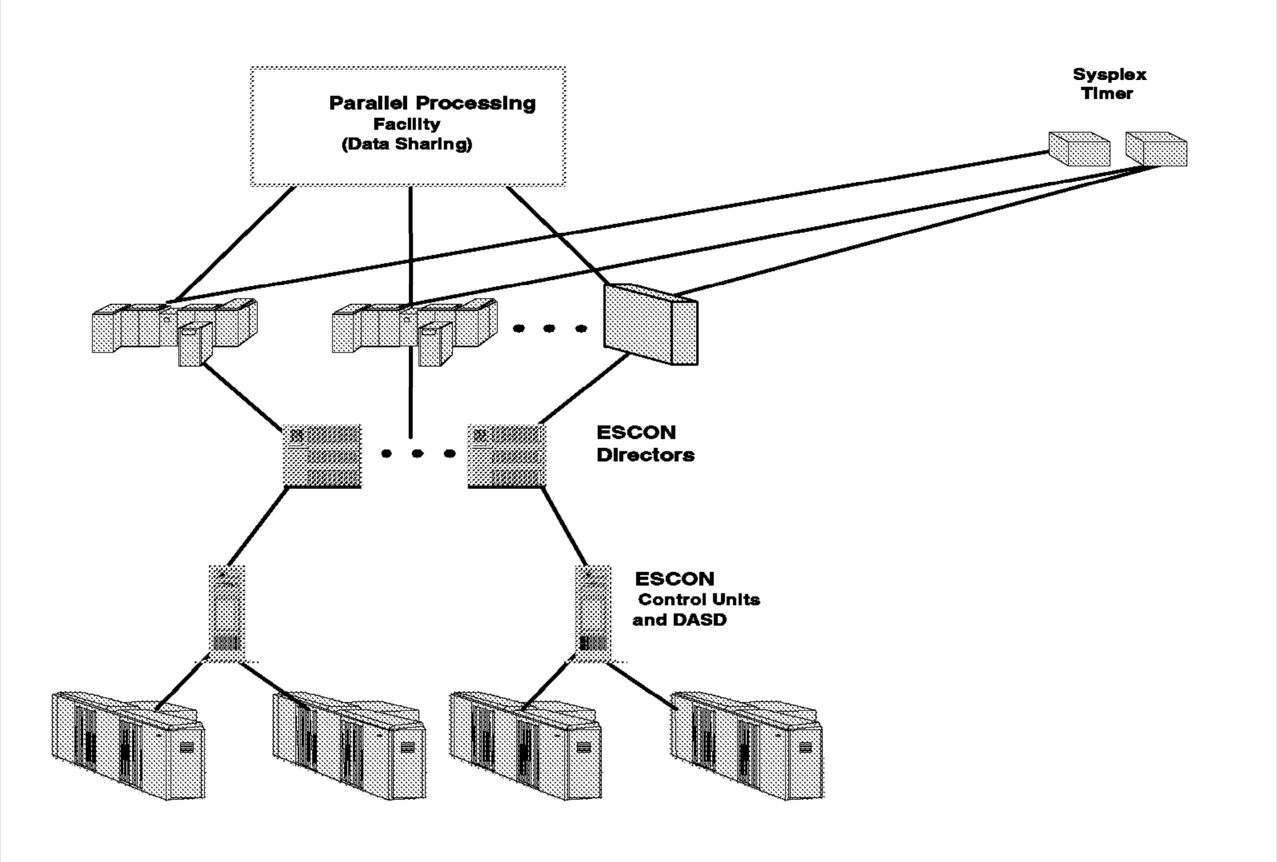


Figure 3. Parallel Sysplex Components

2.1.2 Configuration Scenarios

Several system configurations are possible in an IMS parallel sysplex environment. The configuration you implement will depend largely on the benefits you expect to gain by moving to a sysplex. We explore some of the possible configurations, and the considerations associated with them, although these configurations are by no means exhaustive. First, we define some important terms:

- Affinity
- Vertical partitioning and application split
- Horizontal partitioning and application cloning

Then, we examine the use of MSC within the following configurations:

- Transaction distribution using front-end IMS network interfaces to vertically partitioned sets of applications
- Transaction distribution using front-end IMS network interfaces to horizontally partitioned sets of applications.

2.2 Affinities

When a transaction must run in a specific, unique environment in order for it to run successfully, it has affinity status. We say the transaction has an affinity for that place or device. For example:

- A particular level of a product, such as DB2 V3, or database type, such as Main Storage Data Base (MSDB), or IMS/ESA V5.1 database feature, such as Fast Path Data Entry Databases (DEDBs) with Sequential Dependents (SDEPS) or Virtual Storage Option (VSO) DEDBs, that are not capable of data sharing can only update databases from one particular system.
- Some applications may require access to resources, such as MVS/OS files, communication links, or data retained in virtual memory, that may not be available to all server systems.
- Certain applications may have dependencies on the sequence of a transaction message flow, which might be compromised in a cloned system environment.
- If deadlocks are the result of the merge of certain applications into a data sharing environment, a subset of these applications could be placed in an affinity classification until the bottlenecks are resolved.

Also, from the perspective of transaction response time, transaction throughput, and CPU utilization, it might be advantageous to route certain transactions to a given IMS system for execution. Transactions requiring more resources might be routed to systems with more capacity or to high performance systems if the transaction has a higher business priority.

In sysplex environments using MSC, many typical user transaction flows involve multiple message switches. For example, four transactions are executed within a sysplex with two IMS systems and are routed to each other in the following manner:

TRAN A -> B -> C -> D -> Response to Terminal

In this example assume that transaction B has affinity, transactions A, C, and D do not. The user must decide what to do with transactions C and D: route them back to the originating system or run them on B′s system. This could have unforeseen impacts on workload balancing efforts, depending on typical transaction sequences. A balance exists between the benefits of reducing the MSC flow and the degree of imbalance induced by retaining transaction C and presumably D on the system for which B has affinity.

In our example above, if 100% of the application workload is driven by the transaction A->B->C->D flow and transaction A′s input is equally spread across two IMS systems, where the installation chooses to execute transactions C and D is critical to workload balancing.

You must identify affinities and then decide whether to tolerate or eliminate them. Affinities can be tolerated by ensuring that workloads with affinities run only on a particular subsystem. Eliminate affinities only when the benefit outweighs the cost, as when:

- A small application change increases parallelism and therefore application throughput.

- An application has a critical need for higher availability, and eliminating the affinities enables it to be dynamically routed across several systems.

If applications are rewritten, consider removing affinities at the same time. Also, in future application development, use techniques that avoid affinities.

2.2.1 Identifying Application Affinities

There is unfortunately no easy way of identifying application affinities associated with DL/I bottlenecks for IMS transactions. There are, however, facilities that can be used to identify the DL/I databases that are accessed by IMS transactions and batch message processing (BMP) programs and for those applications requiring more review, there is a method of separating the DL/I calls from the application logic:

- The IMS /DISPLAY command provides information about the databases an IMS transaction or BMP references. Specifically the /DISPLAY DATABASE ALL displays the status of all defined databases, including MSDBs or DEDBs with SDEPs. Then the /DISPLAY DBD dbdname displays, for the data base that is referenced, the type, the PSBs accessing it, and the type of access. /DISPLAY DBD can be used only if Fast Path is installed. This technique can be used to identify the databases accessed by the applications.
- The program specification block (PSB) trace can be run, which records all of the full-function IMS DL/I database calls issued for the traced PSB. For example, to run a PSB trace for PSB FN71, issue the following command:

  ```
  /TRACE SET ON PSB FN71
  ```

  The information resulting from the use of this command is written as a type X′5F′ log record.

  To retrieve the traced data from the log, use the DFSERA10 utility and the DL/I Call Image Capture routine, DFSERA50. Refer to the *IMS/ESA V5 Utilities Reference: System* SC24-8035-00 for information about printing trace records. The output from the DL/I Call Image Capture routine can be used as input to DFSDDLT0; the DL/I test program, where the pattern of DL/I calls to the target databases can be executed repetitively for contention analysis.

2.3 Vertical Partitioning

With a vertical application split, or vertical partitioning, the IMS applications are divided among multiple IMS subsystems (see Figure 4 on page 9). The terminal network is replicated on each IMS system. Users need to know on which IMS subsystems their applications run, and they have to log on to the appropriate subsystems.

Users requiring access to applications A, B, and D are forced to log on to several separate sessions. There would probably be very little data sharing in this configuration. Commonly used databases such as rate tables could be shared among systems.

Determining how to split the applications to different IMS systems is no easy task. If you are coming from a single IMS system, the chances are that your applications are highly interrelated. Vertical partitioning typically requires significant rework of your applications and some database restructuring.

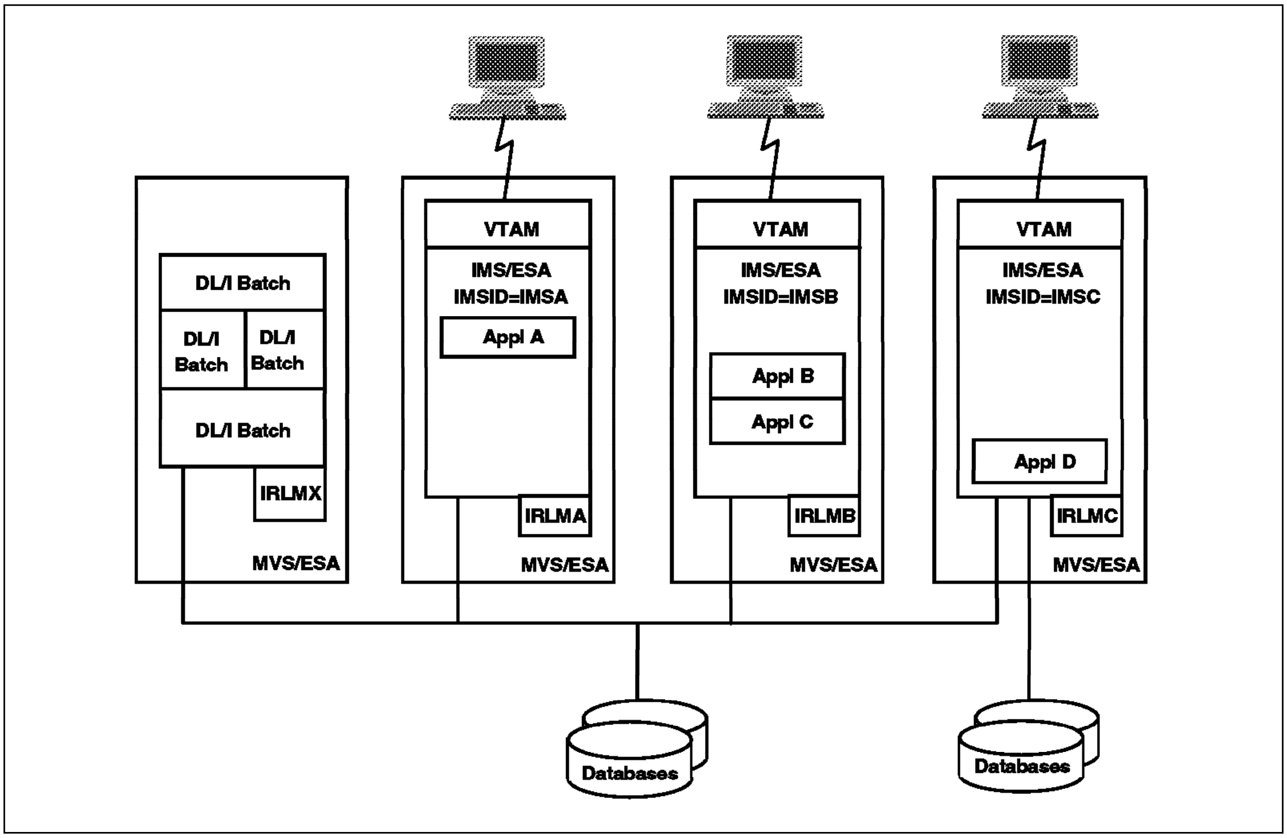


Figure 4. Vertical Partitioning

2.3.1 Advantages

The advantages of vertical partitioning include:

- The systems can be tuned to the specific needs of each application, if required. This approach can be used to bring new applications onto the systems. It may be possible to isolate critical applications from the effects of applications that are not yet stable or tuned by dedicating an IMS system to this workload.
- If one of the IMS systems fails in this environment, other applications are still available on the remaining systems.
- Minimum data sharing would be required, which could lessen the scope of effort for implementation.

2.3.2 Limitations

There are four main limitations:

- Load balancing is not possible.
- Users typically need access to multiple IMS systems.
- Different IMS systems have to be maintained.
- Certain applications will be unavailable if an IMS system fails.

2.4 Horizontal Partitioning

With horizontal partitioning, or horizontal network split, the terminal network is divided among multiple IMS systems (see Figure 5). The IMS applications are replicated, or cloned, on each IMS system, as far as possible. A cloned IMS system configuration is one in which application system resources, including transactions, programs, and databases, are shared among multiple IMS systems operating in a parallel sysplex configuration and sharing a common transaction workload. Transactions executing in such a configuration have access to a common set of DL/I databases through IMS/ESA V5.1 N-Way data sharing support.

In the banking environment, it may be feasible to have some terminals in a branch connected to one IMS, and others connected to another IMS. If one of the sharing participants fails, access to the applications is still possible from a terminal defined to another IMS. The network could be split, for example, by geographical area or time zone. Users have to know to which IMS systems their terminals belong, or, if dynamic ETO terminals are used, to which systems their user IDs are authorized. Once logged on, they have access to all applications to which they are authorized.

Session manager products such as NetView/TAF direct logons according to external criteria. For example, logon requests could be routed according to the alphabetical sequence within the user IDs.

With horizontal partitioning, as far as possible, replicated IMS instances are to run on all images in the sysplex.

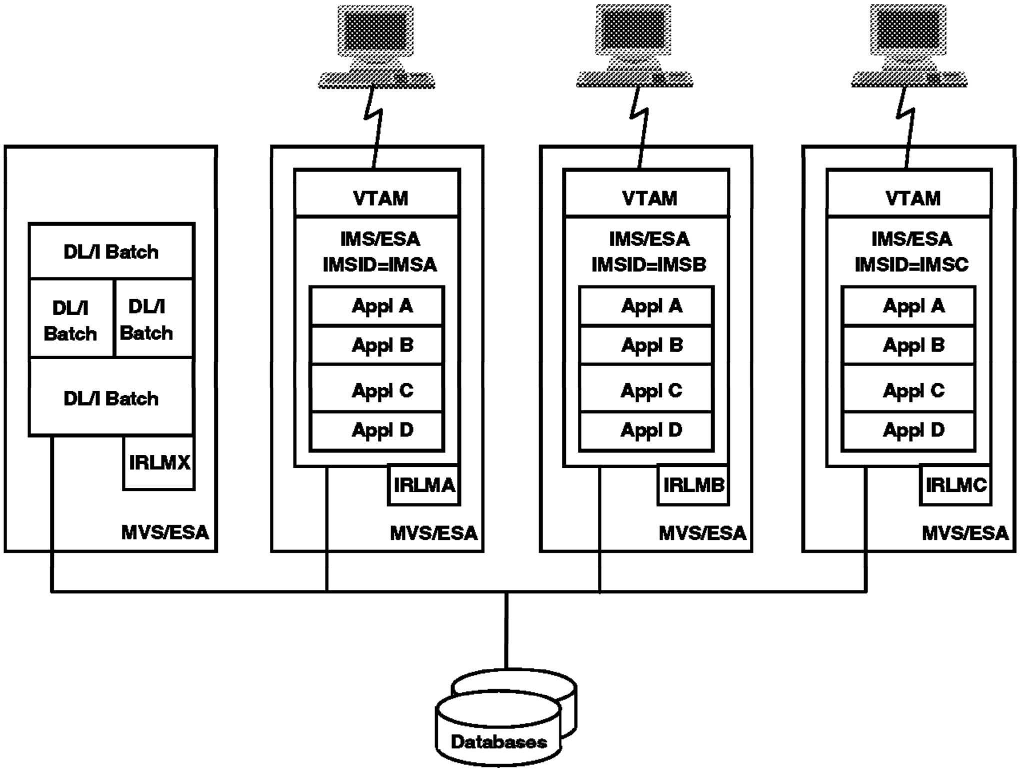


Figure 5. Horizontal Partitioning

2.4.1 Advantages

Horizontal partitioning has the following advantages:

- Cloned IMS systems enable workload distribution in the event of a failure or when an individual system is scheduled for maintenance. The amount of work required by the systems programmer to define and support the environment is also significantly reduced. In addition, using ETO instead of static terminals allows for almost identical IMS systems. Most of the definitions for IMS system generation could be used for all systems.
- A failure of one of the systems would affect only a portion of the network. All applications would still be accessible.
- Load balancing can be done by monitoring and then redistributing the terminals among IMS subsystems.

2.4.2 Limitations

Horizontal partitioning has these limitations:

- Load balancing is possible but could be complex to achieve. Network changes in particular must be controlled.
- Users have to log on to different systems, depending on where their terminals or, if using ETO, their user IDs are located.
- Access to an IMS system must be available for those applications using database types that are unable to data share.

2.5 Multiple Systems Coupling Connection of Partitioned IMS Subsystems

To extend throughput beyond the capacity of a single processor, MSC can be used to connect partitioned IMS subsystems together. Partitioning can be introduced in two ways:

- Vertical partitioning

  Applications execute in one system with their own unshared databases. Some commonly used databases, for example, interest rate tables or a security database, can be shared.

- Horizontal partitioning

  Applications execute in more than one system with shared databases.

In either case, the requirements for the use of MSC still exist: in a vertical partitioned environment, to transfer processing responsibilities among IMS systems, and in a horizontal partitioned environment, to balance the load. The opportunity to use MSC as a front-end switch to multiple, block-level, data-sharing IMS subsystems is discussed in the sections that follow:

2.6 Network Front-Ends, Vertical Partitioning, and MSC

Running the network in a single IMS system (front end) connected by MSC to several IMS systems whose applications are split vertically (see 2.3, “Vertical Partitioning” on page 8) is another configuration approach.

Your transaction can be defined so that no matter which IMS subsystem originates the transaction, it always runs on a particular IMS (see Figure 6 on page 12). Define your MSC links so that each transaction is either local or remote. When a transaction reaches an IMS subsystem, it checks to see whether the transaction is local or remote. If local, IMS processes the transaction; if remote, IMS passes the transaction along to another IMS subsystem across its MSC links. For example, transaction A (TRANA) can be defined to run on IMSA. If you originate TRANA on IMSB, the transaction is processed on IMSA because IMSB knows TRANA is remote and sends it along to IMSA (for which TRANA is local).

Transaction routing from the front-end system to the IMS system where the application is defined can be achieved by using the MSC Routing exits, which are enhanced in IMS/ESA Version 5.

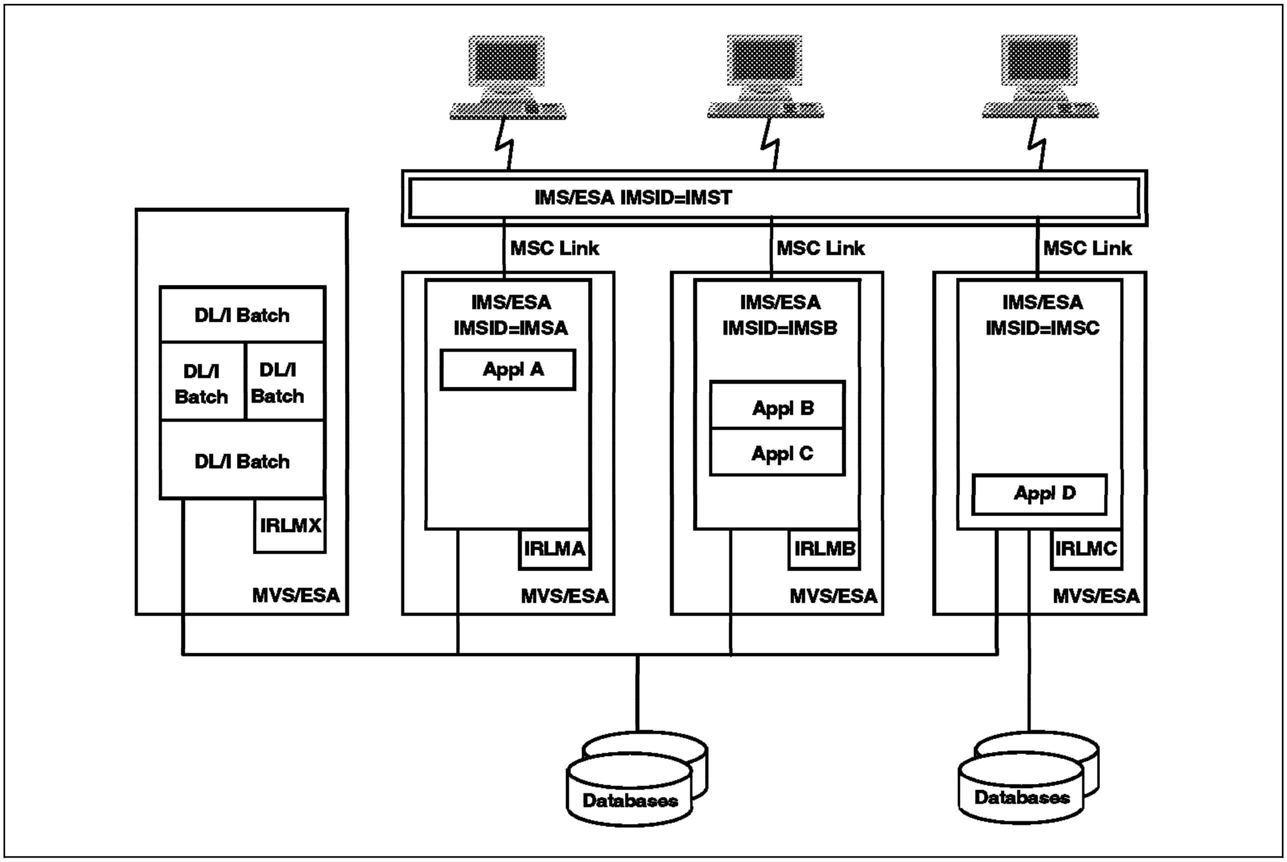


Figure 6. Transaction Distribution with MSC

2.6.1 Advantages

Using vertical partitioning with a network front end and MSC has these advantages:

- You need only log on to a single IMS system (front end) to access to all of your applications. Maintaining signon authority for multiple IMS systems is not required.
- This approach can be used to address the problem of transaction affinities where transactions must run in a particular IMS system in order to run successfully.
- Transactions can be routed to IMS systems where “non-shareable” database resources are defined, such as MSDBs, VSO DEDBs, and DEDBs with SDEPs.

2.6.2 Limitations

Using vertical partitioning with a network front end and MSC has these limitations:

- Multiple, different IMS systems have to be maintained.
- Performance could degrade, depending on the volume of transactions that are routed with MSC logical links. Multiple MSC links must be used together with the Input Message Routing exit (DFSNPRT0) to balance the MSC workload over more than one MSC link.
- Message input received through MSC links cannot be directed to a Fast Path application or passed to the Fast Path Input Exit routine.
- Each transaction has an additional processing overhead (due to the additional processing on the front end).
- Certain applications are unavailable if an IMS subsystem fails.
- All applications will be unavailable if the front end system fails.

2.7 Network Front Ends, Horizontal Partitioning, and MSC

Running the network in a single IMS system (front-end) connected by MSC to several cloned IMS systems is an attractive solution for configuring IMS in a parallel sysplex (see Figure 7 on page 14).

The front-end IMS system is responsible for the correct routing of the transactions to the appropriate cloned IMS system. To make the front-end IMS as stable as possible, no additional workload is run on it. All terminals are attached to the front-end IMS.

Transactions with affinities could be routed to the appropriate IMS system that satisfies their specific requirements. Workload separation could be achieved by allowing certain transactions to run on specific IMS systems through the use of application layered directed routing, user exits, or workload routing products.

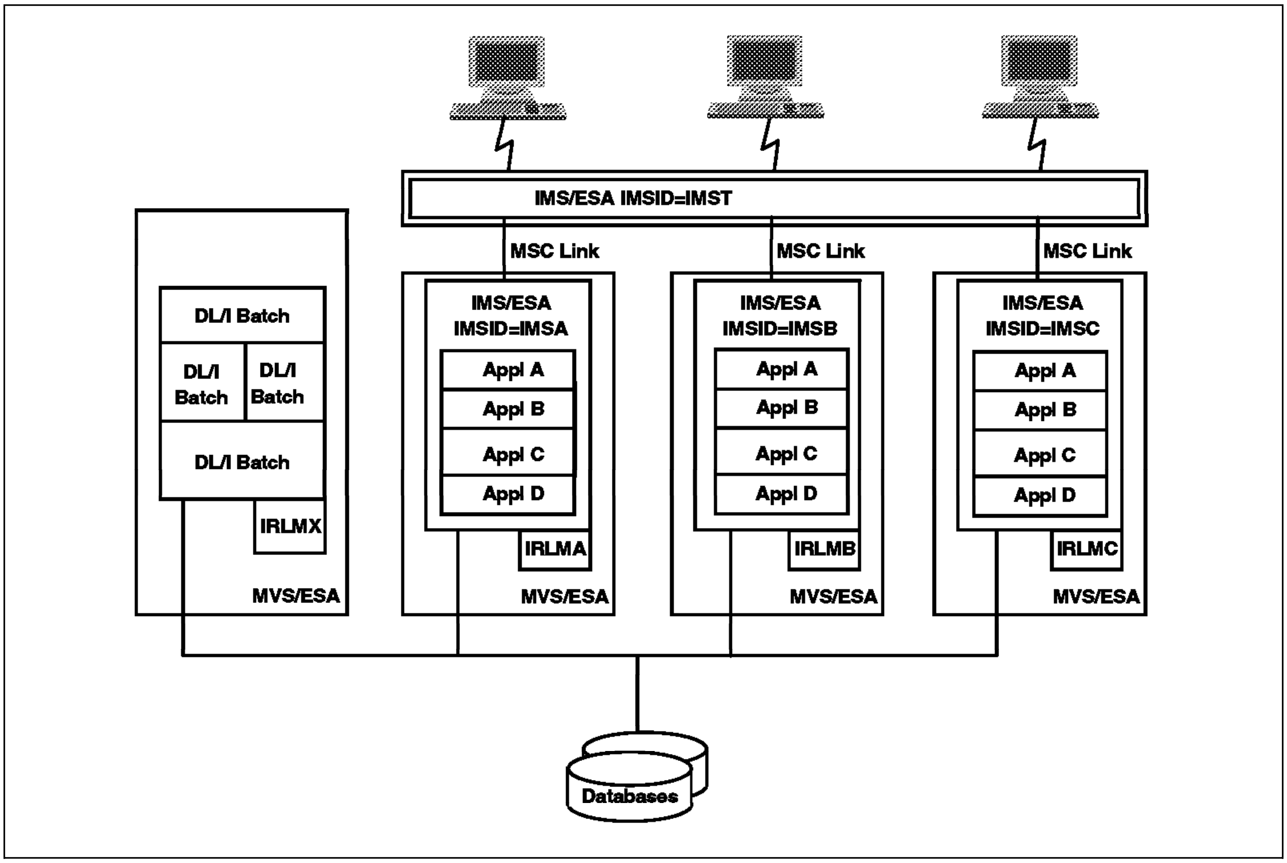


Figure 7. Horizontal Partitioning Using a Front-End IMS with MSC

2.7.1 Advantages

Using horizontal partitioning with a network front-end and MSC has these advantages:

- Because the IMS systems are cloned systems, they are identical in system definition, with such exceptions as the IMS Master Terminal Operator (MTO), IMSID, and MSC definitions. Thus the amount of work required to set up an IMS subsystem environment is significantly reduced. Adding a new IMS system to the sysplex is relatively simple, as it is simply a copy with few definition changes.
- Load balancing could be achieved by using routing algorithms in Input Message Routing exit DFSNPRT0 to control the routing of transactions to the various IMS systems.
- All terminals could be attached to the front-end IMS. If you are not yet using ETO and have a large terminal network, checkpoint time could be drastically reduced on the back-end IMS systems.

2.7.2 Limitations

Using horizontal partitioning with a network front-end and MSC presents these limitations:

- Each transaction has additional processing overhead, because of the additional processing on the front-end IMS system.

- Controls are needed to ensure that workload separation for affinities is enforced where required. Messages received through MSC links cannot be directed to a Fast Path application or passed to the Fast Path Input exit routine.
- Function must be added to DFSNPRT0 for it to determine whether an MSC link is unavailable and, if so, what to do.
- MSC link definitions must be unique in the IMS systems.
- The front-end system is a single point of failure.

Chapter 3. MSC Generation and Processing Overview

In this chapter, we define some relevant MSC terminology, present the Stage 1 macros and parameters used to generate an IMS/ESA MSC system, and discuss several implementation considerations.

3.1 Relevant Terminology

A number of terms have particular meaning when used in an MSC context

- **Links**

  A link represents the connection between two IMS systems. The link has two distinct components, the logical link and the physical link.

  – Physical Link

    The physical link is the actual hardware that connects the IMS systems together. Messages destined for a partner IMS system are sent through a physical link associated with a logical link. For a given active physical link SEND and RECEIVE command flow, communication activity can occur concurrently.

  – Logical Link

    The logical link relates the physical link to the transactions and terminals that use the physical link. In order for two IMS systems to communicate, they must be defined as *partners*, and the logical links must be assigned to the same physical link. Each physical link can have multiple logical links assigned to it. The period of time a logical link is busy determines the capacity of the logical link. In Chapter 6, “Identifying and Solving MSC Problems” on page 41 we describe the events associated with the period a logical link is busy during MSC message routing.

- **Logical Link Path**

  The logical link path identifies the gate to a remote system for a defined set of transactions and logical terminals. It is the path between any two systems; one or more logical link paths must be defined for each logical link.

- **Local or Input System**

  The local or input system is the system that owns the terminal that enters the transaction.

- **Remote or Destination System**

  The remote or destination system is the system to which a transaction can be routed across an MSC link for processing. The transaction is defined as remote by specifying the SYSID parameter in the TRANSACT macro in the system definition. IMS will then pass this transaction across to the remote system. The receiving IMS system should have this transaction defined as local. The transaction will then be processed on this system, and the response returned to the initiating system.

Figure 8 on page 18 shows the relationship of the link components.

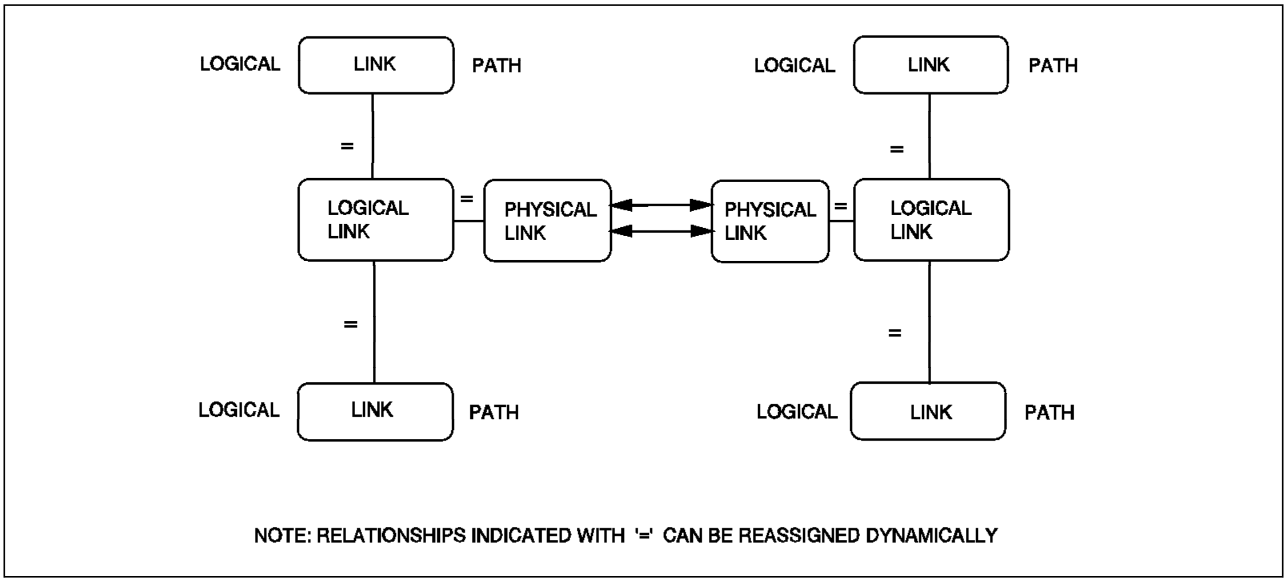


Figure 8. Relationship of Link Components

3.2 Macros and Parameters

Several macros and parameters are required in the IMS system definition to implement MSC. This section describes these macros and parameters and offers some advice on how to implement MSC. The VTAM macro set (if used) must occur after the MSC macro set in the Stage 1 input deck.

- IMSCTRL macro

  An additional parameter has to be added to the IMSCTRL macro for an MSC environment:

  - The MSVID=xxx parameter adds a one- to three-digit decimal number between 1 and 255 as a suffix to the DFSMSxxx control block, used in the multiple-systems verification utility. This suffix is not used anywhere else. Do not confuse it with the PARTNER= parameter.

- MSPLINK macro

  The MSPLINK macro describes the physical link between the systems. The major parameters are:

  - LABEL of MSPLINK macro

    Names the link in order to allow assignment of a logical link to the physical link. The link name is used in the MSPLINK=keyword operand of the MSLINK macro.

  - TYPE

    Depicts one of the four physical links. Valid parameters are BSC, CTC, MTM, and VTAM.

  - ADDR

    Specifies the address of either the communication line or the channel adapter. It is required for BSC and CTC links and invalid for MTM and VTAM link types.

  - BACKUP

    Specifies the control of automatic BTAM or VTAM MSC Link restart after takeover. Use only when HSB=YES (coded for extended recovery facility (XRF) support) is specified on the IMSCTRL macro. The parameter is coded BACKUP=X where *X* is an integer from 1 through 7 that specifies the priority for reestablishing the session. It defaults to 4 when either the keyword or the parameter is omitted. Specifying BACKUP=NO suppresses the session recovery of the MSC physical link at takeover.

  - BUFSIZE

    The size of the input and output buffer for each logical link belonging to this physical link. If two logical links are defined under a physical link specifying BUFSIZE=nn, four buffers (two input and two output) are allocated, each with space=nn. These buffer sizes must be consistent between the partner systems. MSC VTAM links require a minimum buffer size of 208 bytes and allow a maximum size of 30720 bytes.

  - DDNAME

    The ddname of the job control language (JCL) statement that describes this physical connection. It is required for BSC and CTC links and invalid for MTM and VTAM link types.

  - NAME

    Defines the VTAM application ID of the partner system at the other end of the link. It is required only when TYPE=VTAM. The name specified must be the same as the label on the VTAM APPL statement for the remote system (the minor node name) or the remote APPLID startup override name.

  - SESSION

    Defines the number of sessions possible on one physical link. This parameter applies to VTAM links only, and has a default value of 1. A large SESSION value enables the user to dynamically assign more logical links to the physical link than were originally assigned during system definition. The SESSION value can be increased through the use of the NLXB startup parameter.

- MSLINK macro

  Defines the logical link and establishes the partner ID. The major parameters are:

  - BACKUP

    Allows the user to override switching options specified on the MSPLINK macro. It specifies the control of automatic BTAM or VTAM MSC link restart after takeover. Use only when HSB=YES (coded for XRF support) is specified on the IMSCTRL macro. The parameter is coded BACKUP=X, where *X* is an integer from 1 through 7 that specifies the priority for reestablishing the session. It defaults to 4 when either the keyword or the parameter is omitted. Specifying BACKUP=NO suppresses the session recovery of the MSC physical link at takeover.

  - PARTNER

    A two-character identification that must be matched by the other link partner′s system definition. It ensures that the two related logical links in two systems are always logically and physically connected.

  - MSPLINK

    Defines the assignment of the logical link to the physical link. An IMS operation can change this parameter with the /MSASSIGN command. If MSPLINK is not coded at IMS system generation time, a link must be assigned by operator command before any communication can be established between the systems.

  - OPTIONS=SYNCESS or FORCSESS

    SYNCESS stops session initialization if the link partners are not synchronized. This may be the case if one of the partners is cold-started and the message sequence numbers of the partner system indicate a prior session. SYNCESS is the default.

    FORCSESS allows session initialization even if the message sequence numbers are not synchronized.

- MSNAME macro

  Defines the logical link path and establishes a specific multiple system environment by specifying the two SYSIDs (remote and local). MSNAME is an installation-chosen name for the logical link path and defines the link name block that relates to the MSLINK statement and the corresponding remote and local SYSIDs.

  - LABEL of MSNAME macro

    Becomes the name used in IMS commands that concern the logical path, for example,

    /MSASSIGN MSNAME ... or

    /DISPLAY ASSIGNMENT MSNAME ... MSNAME ...

  - SYSID=(rrr,lll)

    Identifies the remote and local SYSIDs for this logical path. All APPLCTN and TRANSACT macros with the same SYSID are therefore assigned to this logical path. The /MSASSIGN command can be used to change the SYSID.

    The SYSID parameter can have values in the 1 through 255 range in IMS Version 5. Selecting low values saves on storage, as control blocks are built according to the highest value.

- NAME macro

  The name of remote LTERMs associated with a physical terminal defined in a remote IMS system. It is only used when:

  - Destination for a message sent by an application program through an insert (ISRT) call to an alternate PCB
  - LTERM to LTERM message switch
  - Target of /BROADCAST LTERM xxxxxx or /FORMAT modname LTERM xxxxxx commands.

  The NAME macro provides the logical terminal name for a physical terminal that is part of the remote system specified in the MSNAME statement. It is the standard NAME macro but is defined under the owning system′s MSNAME.

  If directed routing is used, destinations do not have to be defined in this manner.

  LTERMs receiving replies by ISRT to the IOPCB need ***not*** be defined in this manner.

- APPLCTN macro

  An additional parameter can be specified on the APPLCTN macro:

  - SYSID=(rrr,lll)

    Specifying remote and local SYSIDs on the APPLCTN macro establishes the destination of all transactions linked to this APPLCTN. The values for remote (rrr) and local (lll) are also defined on one of the MSNAME macros.

    Each IMS system must be assigned at least one unique SYSID. The SYSID is local to the owning system and remote to any other IMS system that has a path to this system.

- TRANSACT macro

  The TRANSACT macro also allows for the definition of the MSC SYSID keyword:

  - SYSID=(rrr,lll)

    Specifies the destination of the transaction. The attributes of the local transaction must match the attributes of the remote transaction.

- COMM macro

  The COMM macro is used to specify the general communication requirements that are not associated with any particular terminal type. The COMM macro is always required for terminal types supported by VTAM and is optional for all other terminal types.

  - APPLID

    specifies the application identification in the VTAM access control blocks (ACBs) if VTAM is present. This value can be overridden by specifying APPLIDx= (where *x*=1 to 3) on the IMS startup procedure.

  - OPTIONS: User-written MSC exits

    MSLEXIT, if specified, indicates that the user-written MSC Link Routing exit (DFSCMLR0) is to be included in the generated system.

    MSPEXIT, if specified, indicates that the user written MSC Program Routing exit (DFSCMPR0) is to be included in the generated system. The manual entitled *IMS/ESA Installation Volume 2: System Definition and Tailoring* (SC26-8024-01) indicates under the COMM macro that MSPEXIT is no longer supported. Through APAR PN72686, closed as a documentation APAR, this statement is indicated as being incorrect. The COMM macro MSPEXIT parameter is supported and is required to include the DFSCMPR0 exit.

3.3 Sample MSC System Generation Input

Individual IMS/ESA systems in a sysplex connected by MSC can play several roles. They can act as the input system, the intermediate system responsible for routing transactions, or the destination system in which the target transaction is to be processed.

In the Stage 1 input sample in Table 1, IMSAPROD routes transactions to the remote MSC partner, IMSBPROD. The two systems in the sample system definition are connected by an SDLC link using VTAM with a maximum of five parallel sessions.

Table 1. Portion of IMS/ESA Stage 1 Input for IMSAPROD

Label	Macro
	IMSCTRL ....., MSVID=001
	APPLCTN PSB=A,SYSID=(2,1) or
	TRANSACT CODE=A,SYSID=(2,1)
LINK12	MSPLINK TYPE=VTAM,NAME=IMSBPROD,
	SESSION=5
	MSLINK MSPLINK=LINK12,PARTNER=AB
MSC12	MSNAME SYSID=(2,1)
	NAME LTERMBA
	COMM APPLID=IMSAPROD

Table 2 shows a portion of the IMS/ESA Stage 1 input for IMSBPROD.

Table 2. Portion of IMS/ESA Stage 1 Input for IMSBPROD

Label	Macro
	IMSCTRL ....., MSVID=002
	APPLCTN PSB=A
	TRANSACT CODE=A
LINK21	MSPLINK TYPE=VTAM,NAME=IMSAPROD,
	SESSION=5
	MSLINK MSPLINK=LINK21,PARTNER=AB
MSC12	MSNAME SYSID=(1,2)
	NAME LTERMAB
	COMM APPLID=IMSBPROD

When the input for transaction A arrives on IMSAPROD, MSC ships it to IMSBPROD because it is defined as a remote transaction in IMSAPROD. Because PARTNER=AB is on both MSLINK statements in both systems′ Stage 1 decks, the two logical links in the two systems are connected.

3.4 Implementation Considerations

When implementing an MSC environment, it is vital that you plan and document the design of links between the various systems. In this section we briefly discuss some of the issues to consider when planning for an MSC environment.

3.4.1 Naming Conventions

It is important to establish and document naming standards because the names of the various IMS resources are binding on all participating systems in an MSC environment.

Even if the MSC complex is initially planned as a means of distributing the workload by implementing a front-end or back-end solution, consider that future business needs will require connecting additional systems in a sysplex environment.

In an MSC environment consisting of more than two partners, it may be necessary to supply a range of SYSIDs and partner IDs to each of the participating systems. This will enable you to install new links for additional workload splits in the future.

In establishing naming standards, follow these guidelines:

- Any MSC resource such as a transaction code, logical terminal name, or MSNAME must be unique throughout the MSC complex.
- Transaction codes to be processed in a remote system must reflect such processing in their names.
- The fewer MSC resources you define at system generation, the more easily you will be able to manage the environment and save on computer resources. As the next point explains, however, there may be situations where you should define more resources than are currently required.
- Physical links without a logical link assignment may be defined at system definition time for backup and testing purposes if the resource exists. This increases the storage utilization for control block structures but improves on the options for system diversification for testing and production requirements.
- To allow more than one logical link to be assigned to a physical VTAM link, you must define parallel sessions on the MSPLINK macro. The logical link must be assigned to a physical link before communication can be established. This assignment can be done at system generation time or with the /MSASSIGN LINK X TO MSPLINK Y command on both systems′ master terminals.

3.4.2 Network Definition Considerations

When planning for the network, keep in mind that message queues for an input, intermediate, or remote system must allow for the remote transactions to be queued. The message lengths and their expected loads must be taken into account when allocating space for the message queues.

In addition to resources used to transmit the transaction over physical links to the remote processor, resources are needed for message queuing and logging. For example, in IMS/ESA Version 5 the prefix of each message has a 64-byte MSC system segment item and a 40-byte MSC extension item added to it. The prefix size of the LU6.2 message is increased to a minimum value of 480 bytes when these messages are shipped across MSC links. Therefore, link buffers defined by the BUFSIZE parameter on the MSPLINK Stage 1 macro must be large enough to hold these messages. The size depends on the total message traffic queuing estimates.

Although MSC physical and logical links continue to be predefined through IMS system definition, you can use ETO to dynamically create MSC remote LTERMs during IMS initialization through MSC descriptors. The descriptor relates each remote resource with the link path name of a generated MSNAME macro. The descriptors are located in IMS.PROCLIB members DFSDSCMx and DFSDSCTy. DFSDSCTy contains the installation generated descriptors to avoid loss when member DFSDSCMx is replaced by a subsequent IMS generation. The x suffix for DFSDSCMx is the IMS suffix. The y suffix for DFSDSCTy is selected for installation-generated members and is specified as a control region parameter.

When planning for network availability, remember that MSC does not increase the availability of a single IMS system, but it does help increase the availability of your overall IMS network.

Chapter 4. MSC Operations

In this chapter we discuss the various commands used to operate and monitor MSC links and present an overview of security in an MSC environment. We do not provide a reference to each command and operand. Instead, we present some examples of the command flow necessary to dynamically modify the assignment and status of links and other MSC-related resources. Sysplex configurations may require a wider use of these commands to balance and control workload flow, and to optimize system performance. For more information, refer to:

- *IMS/ESA V5 Administration Guide: Transaction Manager*, SC26-8014-00
- *IMS/ESA V5 Operations Guide*, SC26-8029-00
- *IMS/ESA V5 Operator's Reference*, SC26-8030-00

Each system in a multisystem configuration is operationally an independent unit. It exclusively owns its communication resources, which are controlled by its own master terminal. Startup and local operation do not depend on the availability of the physical link.

4.1 Multisystem Communication Initialization

The initial status of an MSC link after any type of restart is PSTOPPED. To activate a session with a remote system partner, issue:

```
/RSTART LINK X
```

where *X* is the name of the MSC link.

With VTAM links, one /RSTART command on either side of the link is sufficient to start communications, but for BSC, MTM, and CTC links, both sides must enter the /RSTART command for the same link. If a link is not successfully started, messages wait until the links have been reassigned.

If a system is cold-started, any queued messages to or from terminals or programs on other systems are lost. The addition of other MSC-connected IMS systems with sysplex configurations increases the impact that a cold start on any one IMS system has on message availability within the sysplex.

In IMS/ESA Version 5 some combination of the /PSTOP LINK and /DEQUEUE MSNAME with additional options is usually effective in releasing an MSC link that is hung.

4.2 Reassigning Logical Links

MSC uses physical and logical links. The physical link, as the physical connection to the partner IMS/ESA system, is used by the logical link. The logical link contains transaction and terminal routing information and is assigned to a physical link by definition at system generation time. The logical link can be reassigned to another physical link through operator commands after it is placed in a stopped and idle status. The following command stream performs the reassignment:

1. **/DISPLAY ASSIGNMENT MSPLINK** *msplinkname* to inquire about the initial setup.
2. **/PSTOP LINK X**, where *X* is the logical link number

   Use the FORCE keyword with the /PSTOP command for VTAM links when an MSC VTAM link will not clean up and return to idle state during normal PSTOP processing, even though VTAM has terminated the session.

   Use the PURGE keyword only when there is one logical link whose physical link is channel to channel. Use it when the partner link is in a system that has failed, but the link does not return to idle status, even though it has been stopped.
3. **/DISPLAY LINK X**, where *X* is the logical link number. The logical link is in PSTOPPED IDLE status.
4. **/MSASSIGN LINK X TO MSPLINK Y** on both IMS/ESA master terminals.

   The input system has a logical link connection to physical link Y for all SYSIDs assigned to logical link X. The /MSASSIGN command alters the assignments of the following multisystem resources: LINK, MSPLINK, MSNAME, and SYSID.
5. **/RSTART**
   LINK X For any link except VTAM, the /RSTART command must be issued on both sides of the link.

4.3 Reassigning Logical Link Paths (MSNAME)

Operational requirements may dictate that an MSNAME be assigned to a different logical link.

A logical link path or MSNAME is assigned to a specific logical link (whether explicitly defined during system generation or by operator command /MSASSIGN):

1. Use the /DISPLAY command to identify the assignment of the MSNAME.

   /DISPLAY ASSIGNMENT LINK ALL

   The command results in the following sample output:

   ```
   LINK PLINK SIDR SIDL MSNAME
    1   LINK1
    2   LINK2  42   43  IMSTST
   ```

   The logical link path IMSTST is connected to logical LINK2, which in turn is assigned to physical LINK2.
2. The logical link may not transmit during reassignments, so it must be stopped and idle before entering the following:

   /MSASSIGN MSNAME IMSTST TO LINK 1
3. Issue **/RSTART LINK 1** to enable the message flow through logical link path IMSTST.

4.4 Reassigning Transactions

Transactions can be reassigned from a local to a remote system or the reverse. This reassignment might become necessary to balance the sysplex workload or adjust affinities. Such a reassignment is feasible only if the specific transaction is initially cloned in the sysplex partners where the reassignment is to take place. In the following example, transaction TESTA is initially local to IMSA and is routed to IMSA if entered on IMSB.

1. **/DISPLAY TRAN TESTA** entered on IMSA creates this sample output:

   ```
   TRAN  CLS  ENQCT ...
   TESTA  1     0 ...
   ```

2. **/DISPLAY TRAN TESTA** entered on IMSB displays:

   ```
   TRAN  CLS  ENQCT ...
   TESTA RMT    0 ...
   ```

3. **/STOP TRANSACTION TESTA** on IMSA and IMSB
4. **/MSASSIGN TRAN TESTA MSNAME** *msname* on IMSA

   AND

5. **/MSASSIGN TRAN TESTA LOCAL** on IMSB

A display on IMSA shows:

```
TRAN  CLS ENQCT ...
TESTA RMT ...
```

After the /START TRANSACTION TESTA command is used, transaction TESTA is executed on IMSB. This reassignment remains active until the next cold start.

4.5 Session Restart

Both MSC and VTAM monitor every message sent over a SNA link by means of sequence numbers. For every message sent, the receiving system replies with an acknowledgment. A missing acknowledgment could point to a link failure or a system breakdown of the partner.

When the link is restarted, the link partners exchange their individual sequence numbers through set and test sequence number (STSN) logic and decide on resynchronization actions:

- If the destination system did not receive the message, it is resent by the input system.
- In case of a successful transmission but missing acknowledgment, the message is dequeued in the input system.

If one of the link partners had to perform a cold start or a restart to an earlier checkpoint, session initiation can be forced by a system definition parameter or an operator command. The system generation value is FORCSESS in the OPTIONS parameter in the MSLINK macro. FORCSESS forces the session to be completed whether or not session resynchronization is successful. Successful session resynchronization occurs when the message sequence numbers of the two logical units in session agree, or when the sequence number of the sender is no more than 1 higher than the sequence number of the receiver. The default is SYNCSESS. The OPTIONS parameter values in the MSLINK macro must be consistent with the MSLINK OPTIONS parameter values in the partner system. During IMS execution, the value of this synchronization option may be overridden by the /CHANGE command.

4.6 MSC Forced Closure of Links

In certain error situations, for example, a lost VTAM, an MSC link could become unusable. With releases of IMS/ESA before V5.1, no IMS command (/CLSDST, /IDLE, /PSTOP) or VTAM command (VARY NET,INACT,FORCE) would rectify the situation. The only solution in IMS Version 4 was to restart the IMS system. The new facility in IMS Version 5 enables the IMS service to continue uninterrupted during the resolution of link problems.

A new option, FORCE, on the IMS /PSTOP LINK command has been introduced in IMS/ESA Version 5 to clean up the link control blocks and so enable the link to be restarted without an IMS restart:

/PSTOP LINK *link #* **FORCE**

When this command is issued, IMS checks that the link is a VTAM link in PSTOP Pending state (PSTOPPED NOTIDLE) and that the session has been terminated. If so, the control blocks are cleaned up and the link status is set to PSTOPPED IDLE.

If the command is ineffective, it is probably because the session has not been terminated. The operator must first identify the session and terminate it, using the appropriate VTAM commands:

D NET,SESSION,LU1=applid1,LU2=applid2,SCOPE=ALL,LIST=ALL

V NET,SESSION,SID=sess-id,NOTIFY=YES,SCOPE=ALL,TYPE=FORCE

It may then be necessary to reissue the /PSTOP LINK FORCE command. After that, issue:

/DEQUEUE MSNAME *msname* **PURGE** or **PURGE1**

to cancel the output message currently being sent on the MSC link. The PURGE option cancels all output messages, but PURGE1 cancels only the first message queued.

4.7 Multisystem Verification

During system definition of an MSC-capable system, a load module is placed in the IMS.MODBLKS data set, and can be used as input for the MSC Verification utility (DFSUMSV0). Executing the batch utility is feasible, however, only if *all* load modules from all IMS systems in the sysplex are accessible. After all system definitions are complete in the sysplex configuration, a link-edit of all multisystem control blocks from all IMS multisystem definitions into IMS.RESLIB or some other user-specified library would have to be completed. Then the utility has access to all the MSC sysplex control block structures. For a large, dynamic sysplex environment, it would be difficult to use the MSC Verification utility to obtain current, accurate status because of this restriction.

The /MSVERIFY command displays the actual status of the IMS/ESA system at the moment the command is processed. The output of this command shows the discrepancies between the local and target remote system definitions.

The /MSVERIFY command validates that:

- Logical terminals exist for the remote LTERMs defined.
- Transactions are defined and have the same attributes in their remote system as in the local system.
- Logical path definitions are consistent and usable.

Verification occurs in two stages. First, the local definitions are gathered and sent over the link to the remote system, where they are compared. This in turn triggers the remote system to send its definitions to the local system. The format of the command is:

/MSVERIFY MSNAME *msname*

or

/MSVERIFY SYSID *sysid #*

Only one remote system can be specified for each /MSVERIFY command. Also, the /MSVERIFY command cannot detect errors caused by improper use of MSC directed routing. Because the link traffic generated through /MSVERIFY could be considerable, the command should be executed judiciously in production environments.

4.8 Security in an MSC Environment

The MSC environment appears as a single system to the end user. In reality, however, it consists of two or more IMS control regions with all security aspects closely connected to each of the IMS/ESA systems. Security maintenance in a sysplex environment is performed independently for each system.

By definition, most of the transaction authorization and password checking takes place in the input system, because most of the remote systems do not have the privileges granted to the user or terminal entering the message.

For directed routing, the back-end system does the security check. The front end system cannot do such a check as the resource is not defined there.

4.8.1 Security Maintenance Utility (SMU)

In general, it is good policy to do as much security checking as possible on the front-end system. SMU can be used for this purpose. If required, SMU can be used in conjunction with RACF.

Signon verification, combined with transaction authorization and password checking, enables the user to control the processing at input time. Password security is verified on terminal input after execution of the Terminal Routing exit (DFSCMTR0). Terminal security is verified on terminal input, and after an application′s use of a DL/I change call, if the call is issued in the input system.

It is possible to achieve the same level of security provided by the combination of SMU and RACF with the IMS/ESA signon verification (DFSCSGN0) and resource access security (DFSISIS0) exits and SMU.

Security maintenance must be performed independently on each of the IMS systems in the MSC environment. Remote transaction names must be declared in the SMU, even if the transaction does not perform locally.

Security checking for transactions being processed in the remote system is also based on the relationship of TERMINAL and TRANSACT SMU control statements, with the exception that the logical link path (MSNAME) has to be defined instead of the LTERM. When a destination system receives a message, it verifies security according to the input terminal′s association with the logical link path. This information is obtained from the type X′01′ input log record created by the remote system that has the MSNAME associated with the MSC link in the field location normally assigned for the LTERM name. With this link security, authorizations for each link path name are defined in the SMU of the destination system. Thus transactions from unauthorized systems cannot be processed.

Intermediate systems do not perform security checking.

4.8.2 Resource Access Control Facility (RACF)

RACF is an IBM-licensed program used with MVS for security. To be used with IMS/ESA on static terminals, RACF requires SMU for resource access security. The RACF security variables, terminal user security, and resource class are not supplied by SMU and the IMS/ESA exit routines.

Most of the above security considerations also apply to RACF-based security. For transaction authorization checking:

- During control region initialization, RACF builds transaction profiles that are checked against a user′s privilege. These profiles can be altered by the online change process.
- During /SIGN command processing, RACF builds profiles representing the user and the user′s privilege.
- During transaction authorization, RACF compares the user′s privilege with the transaction profile and returns a message of authorized or unauthorized to IMS/ESA.

4.8.3 Use of Security Reverification Exit DFSCTSE0

The DFSCTSE0 exit is an entry point within DFSCTRN0, the Transaction Authorization exit routine. DFSCTSE0 enables users to reevaluate their transaction authorization checking on DL/I CHNG calls from both terminal-owning and remote systems. By coding this exit routine, you can avoid a security failure that occurs when a RACF or non-RACF security environment is called in a destination MSC system by a user that is not signed on to that particular IMS system.

To understand the need for DFSCTSE0, compare the local to remote MSC environment where DL/I CHNG calls are being issued.

During /SIGN command processing, tables representing the user and accompanying authorizations are built in the local system. If the transaction issues a CHNG call to another transaction in the local system, the user′s authority to process the transaction is checked against the tables built during signon.

A control block structure called a *communications terminal block* (CTB) is built by the terminal-owning IMS system (local). The terminal-owning system has the physical terminal definitions for the LTERM name that is present in the I/O PCB.

When the transaction is executed in a remote MSC system, a security check for the user cannot be made because the user has not signed on to the remote system. As a result:

- When processing this call on a remote MSC system, IMS cannot verify the user′s security profile. Instead it requests RACF to determine authorization by using the user ID associated with the remote MSC system′s control region instead of the user ID in the I/O PCB.
- Transaction Authorization exit DFSCTRN0 is not invoked unless all remote transactions have been authorized to the user ID of the IMS remote control region (which poses an unacceptable security risk to many MSC installations). In that case, RACF responds with an “authorized” message and exit DFSCTRN0 is called.
- If the above circumvention is not used, RACF on the remote system returns a message of “not authorized.” The Transaction Authorization exit is not called and the application receives an A4 status code for the CHNG call.

DFSCTSE0 is called regardless of the RACF or DFSCTRN0 return codes encountered if it has been coded as an entry point with DFSCTRN0. This entry point in DFSCTRN0 is used in conjunction with RACF or non-RACF security environments. The exit either returns with a Register 15 value of 0, indicating acceptance of the CHNG call, or a value of 8, to inform IMS to present an A4 status code to the application.

Chapter 5. MSC User Exits and Directed Routing

With the advent of the parallel sysplex, some customers may need to configure a greater number of connected IMS systems and may choose to have a basic system definition from which all other systems are cloned. Although the MSC link definitions must be uniquely defined in each IMS system, an objective of some of the IMS 5.1 MSC enhancements is to reduce the need for unique transaction and LTERM names in each system.

Directed routing is an existing MSC facility that can reduce the need for unique names across all systems, but it requires user applications to participate in the routing process.

5.1 MSC User Exits

Several MSC user exits can assist with the routing of messages in a parallel sysplex environment. Figure 9 illustrates available MSC user exits. DFSCMPR0 can be called from local as well as remote systems at the IMS/ESA V5.1 level. The figure has DFSCMPR0 executing only in the remote IMS system.

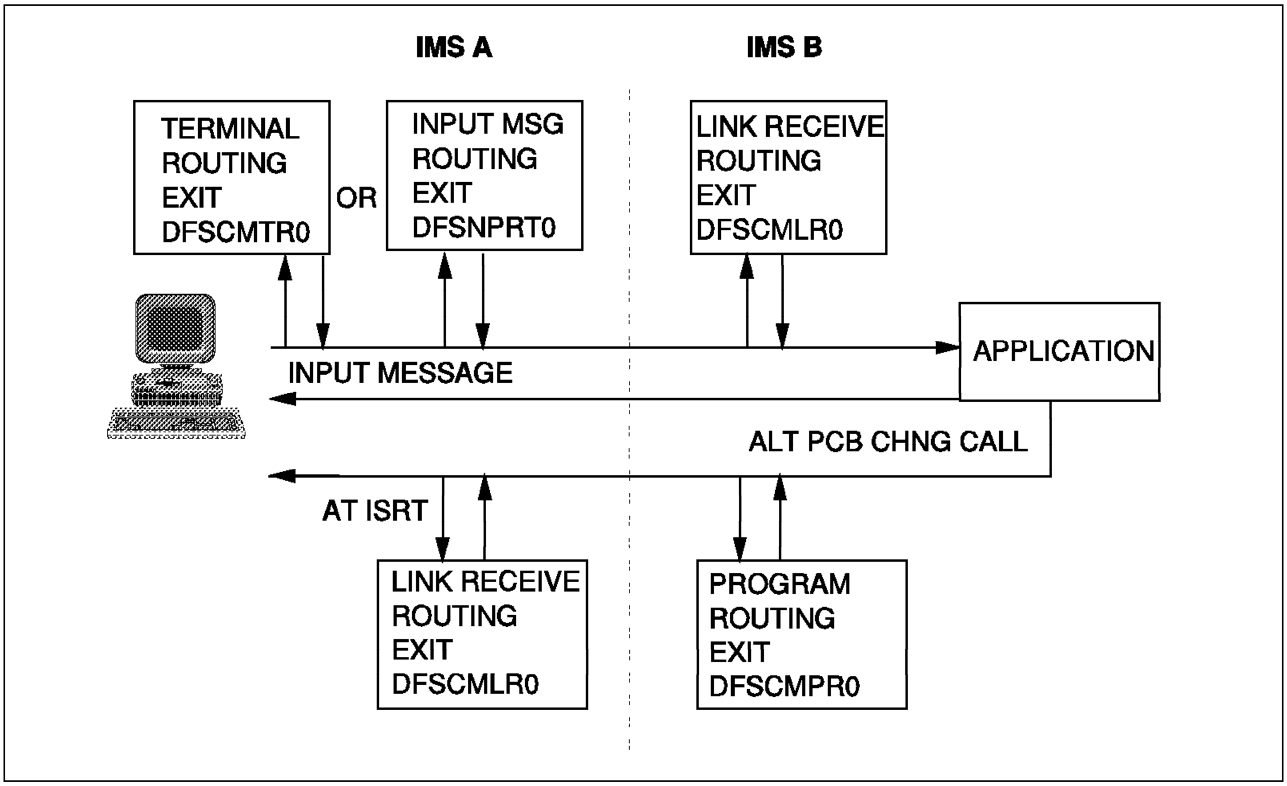


Figure 9. MSC User Exit Positioning in the MSC Processing Cycle

5.1.1 Terminal Routing Exit (DFSCMTR0)

DFSCMTR0 is called when a message arrives from a terminal. This exit is invoked on the input system only. It is called by the communication analyzer before the message is placed in the message queue. It can set the destination to any local or remote transaction known to the originating IMS system. It also provides the option of changing the contents of the message:

- The first segment of the input message is presented to the exit. Any part of the message segment can be changed, except for the length (LLZZ).
- A pointer to the destination of the message is passed to the routine. The pointer itself can be altered to point to another field containing a new destination, or the contents of the field (addressed by the pointer) can be changed.

DFSCMTRO is no longer linked in with the IMS nucleus. IMS issues a load request from the RESLIB concatenation at startup time. If DFSCMTR0 is found, it is enabled.

DFSCMTR0 is not called in the following instances:

- IMS/ESA operator commands
- Message, entered from a terminal in preset mode
- Message, continuing a conversation

The MSC Terminal Routing exit (DFSCMTR0) can be used, but only when the Input Message Routing exit DFSNPRT0 is not used in IMS/ESA Version 5.

5.1.2 Link Receive Routing Exit (DFSCMLR0)

DFSCMLR0 is called when a message arrives from a remote system through MSC. This exit runs in the destination system and has two entry points. Its main entry point, DFSCMLR0, is called by the MSC communications manager. Entry point DFSCMLR1 is called when the first segment of a message from a remote system sent by directed routing with an LTERM destination is received.

On the basis of the current destination transaction and the actual transaction code in the message, DFSCMLRO can set a different destination transaction code.

The following information is passed to DFSCMLR0:

- The address of the first segment of the message in edited format. It also could be the scratch pad area (SPA) in the case of a conversational transaction.
- A pointer to the local transaction code

If the exit leaves the destination unchanged, it has to pass a return code of 0 to the MSC communications analyzer.

If the destination is changed and the area pointed to by Register 2 contains a new transaction code, a return code of 4 is passed back.

Return codes 8 and 12 signal that the transaction code was rejected altogether and that either message DFS2175 or a user message is to be issued. These two options can be used if the exit routine is adapted to inspect the availability of the local destination and to discard the message if the destination is unavailable.

Entry point DFSCMLR1 receives the following information on entry:

- The first segment of the message in edited format
- A pointer to a field containing the destination name
- A pointer to the originating LTERM

The link receive routing exit can change the destination name, but not the pointer to the field. It also must signal its actions via a return code of 0 or 4, indicating unchanged or changed destinations respectively, to the MSC communications analyzer.

5.1.3 Program Routing Exit (DFSCMPR0)

DFSCMPR0 is enhanced in IMS/ESA V5.1 to be called on local as well as remote systems, and several more routing options are available.

The exit is called when an application issues a DL/I CHNG call. Before IMS V5.1, DFSCMPR0 exit could only reroute LTERM destination messages from an application program running in a remote IMS system to an LTERM in the local (input terminal) system, and its options were:

- Let the processing system interpret the CHNG call destination.
- Route the message to the originating system for destination determination.
- Reject the CHNG call with an A1 status code.

In IMS/ESA Version 5, the program routing exit is called in different circumstances, and has a larger range of routing options. It is called whenever a DL/I CHNG call to an alternate modifiable PCB is issued, which is done when the following conditions apply:

- MSC is generated in the system (at least one MSPLINK macro).
- The application is a non-conversational message processing program (MPP) or BMP (local or remote).
- A get unique (GU) against the input/output program communication block (IOPCB) has successfully completed within an application. This includes BMP as well as MPP application programs.
- The CHNG call PCB is not an alternate response PCB.

DFSCMPR0 can return to IMS a SYSID of any IMS system connected to the application program processing system by MSC links and route the message to that system for processing. For messages whose destinations are transactions (program-to-program switches), the exit can indicate which IMS system should process the messages and therefore be used to balance message processing across the IMS systems in the sysplex environment.

The exit recognizes whether the CHNG call is being processed on the originating system by the value in register 0:

- The value of the SYSID in register 0 is positive if the exit is called by a remote MSC system.
- The value of the SYSID in register 0 is negative if the exit is called in the local LTERM system.

The routing function is determined by setting the return code in register 15 as shown in Table 3 on page 36.

Table 3. MSC Program Routing Exit: Routing Options Set by R15

R15	Meaning
0	Use the destination name as supplied by the CHNG call to find the destination.
4	Use the originating SYSID in R0 to route the message back to the specified LTERM in the originating system.
8	Reject CHNG call and return A1 status code.
C	Use the originating SYSID in R0 to route the message back to the specified TRANCODE in the originating system.
10	Use the SYS ID returned in R0, or the MSNAME pointed to by R1, to route the message to the specified LTERM.
14	Use the SYS ID returned in R0, or the MSNAME pointed to by R1, to route the message to the specified TRANCODE.

The way in which DFSCMPR0 is defined and loaded has not changed. The COMM macro OPTIONS=MSPEXIT is still required, and the module is still linked with the IMS nucleus.

DFSCMPR0 from a previous release will almost certainly have to be modified to work with IMS/ESA 5.1 because it is called for all CHNG calls (except in Fast Path EMH programs), not just those on remote systems. Update the exit to check the value of register 0; if it is negative, set RC=0 register 15.

5.1.4 Input Message Routing Exit (DFSNPRT0)

DFSNPRT0 is a new general-purpose input routing exit. DFSNPRT0 is intended to eventually replace DFSCMTR0. Consequently, customers currently using DFSCMTR0 should plan to migrate to DFSNPRT0. Customers expecting to use MSC with LU6.2 or open transaction manager (OTMA) must convert if they use the DFSCMTR0 exit. The old exit can still be used, but only if DFSNPRT0 is not used.

In a sysplex, or other environments where transaction macros are cloned as local transactions across IMS systems, DFSNPRTO can control the routing of transaction messages to other IMS systems. The exit is not specified at IMS generation time on the COMM macro, and it does not require MSC to be specified in the IMS system. Instead, IMS tries to load the exit at startup time from RESLIB or a library concatenated to RESLIB. If the exit is found, it is used.

DFSNPRT0 can specify the destination system, using either a defined SYSID or MSNAME. Thus transactions defined as local or remote can be rerouted to different IMS systems for processing. When rerouting a transaction type message to a remote MSNAME or SYSID, the transaction must be defined in both the local IMS system where the exit routine is being called and the remote destination system to which the transaction message is being rerouted.

DFSNPRT0 is called after receiving messages from the following programs, devices, or clients:

- Any IMS-supported terminal
- Any ISC LU6.1 device or program
- Any Advanced Program-to-Program Communication (APPC) LU6.2 device or program. DFSNPRT0 is always called for input messages from any APPC LU6.2 device or program, regardless of whether or not it is destined for Common Programming Interface-Communications (CPI-C) driven application

programs. More precisely, the exit is called when the APPC conversation is allocated and before data is received.

- Any open transaction manager access (OTMA) client

For all other input messages from IMS-supported programs and devices, DFSNPRT0 is called when the first segment of the message is received from the device or program.

For conversation transactions, DFSNPRT0 is called when the conversation is started (transaction is first received). If the exit reroutes the conversation to a remote system, all continuation messages from the input terminal to the transaction bypass the exit and are routed to the same remote system. This includes continuation messages routed to a different transaction by a deferred program-to-program switch.

Fast Path input transactions are not supported (the exit is not called).

A sample DFSNPRT0 can be found in the IMS TMSOURCE library.

5.2 Using MSC User Exits in a Sysplex

In a parallel sysplex environment, it may be necessary to route certain transactions to a particular system. The enhancements in IMS/ESA V5.1 make this possible without requiring definitions of remote transactions and without changing applications to use directed routing.

5.2.1 Reduction of System Generated Resources

The Terminal Routing exit (DFSCMTR0) or the Input Message Routing exit (DFSNPRT0) can be adapted to direct transactions that have different transaction codes but identical characteristics to one destination and thereby reduce the number of remote transaction codes defined to the local system. By means of tables accessible to the user exits, the desired transaction code can be inserted before being sent from the front-end IMS system. Then the generic transaction code can be used, and a second exit replaces the changed transaction code with the original code, once the message has been received by the remote system. This can be done with the Link Receive Routing exit (DFSCMLR0).

5.2.2 Link Availability Checking

DFSCMTR0 or DFSNPRT0 can be used to inspect the availability of all resources belonging to a logical link path and make rerouting decisions on the basis of the information gathered. One of the parameters passed to the exit is the destination name of the message entered from a local terminal. This destination may be a transaction name or an LTERM name. Transactions with this name can be used to find the relevant control blocks that begin a chain of structures leading eventually to what is called the *logical link block* (LLB), which. represents the logical MSC link. The LLB contains flags indicating the status of the link and can be examined to decide on the routing actions to take. See Appendix A, “Sample Link Availability Check Subroutine” on page 87 for an example of a program that can perform a link availability check.

5.2.3 Load Balancing in a Horizontal Partitioned Environment

Use of DFSNPRT0 facilitates transaction balancing in an MSC environment. For instance, if IMSA had MSC connections to IMSB and IMSC, and TRAN1 was defined as local in all three systems, each TRAN1 message received in IMSA can be either processed locally in IMSA (not rerouted) or rerouted to IMSB or IMSC message by message. In effect all transactions can be cloned, defined as local transactions in each system, and the exit can determine where to route each occurrence of a transaction. The exit could be implemented to recognize when, for example, the remote transaction is unavailable and take appropriate action. The following transaction characteristics must be defined identically in the inputting IMS system and the rerouting IMS system:

```
MSGTYPE = MULTISEG/SINGLSEG
INQUIRY = YES/NO, RECOVER/NORECOV
SPA = (SIZE)
```

An example of the routing parameters used for load balancing is the size of the queue of messages to a particular link. The size can be determined by examining fields CNTNQCT (count of enqueued messages) and CNTDQCT (count of dequeued messages). A decision could be made to choose another link if the tested link has a queue of traffic to transfer. Weighting factors can be assigned to target processors on the basis of CPU power so that a smaller processing unit is not overloaded by routed messages.

5.3 Directed Routing

MSC directed routing enables application program to specify the MSNAME of the system and destination within that system for a message to an LTERM or application program. The receiving application program can determine the MSNAME of the system that originally scheduled it. With directed routing, the specified remote destination, a transaction or an LTERM in another system, does not have to be declared explicitly in the IMS system definition for the sending system.

Changes to application program coding have to be made to introduce MSC directed routing. First a CHNG call is issued against an alternate PCB, supplying the name of the MSC link that connects the two IMS systems. For example, to send a message to TERMINAL 1 in IMSB after receiving a message from some other terminal in IMSA, issue this CHNG call:

CHNG altpcb, LINK1 where LINK1 is the MSNAME.

Then issue an ISRT call (or calls) to send the message just as you would send a message to a local terminal. If your message encounters a security violation, MSC detects it in the receiving system (in this case, IMSB) and reports it to the originating terminal in IMSA.

When directed routing is used, you can specify ROUTING=YES on the TRANSACT macro during system definition to specify whether or not the application program processing a transaction is to be informed of the system that originated the transaction. If ROUTING=YES is specified, IMS TM takes these actions to indicate to the program that the message is from a terminal in another IMS TM system:

- Instead of placing the LTERM name in the first field of the IOPCB, IMS places in this field the MSNAME that corresponds to a logical link path to the originating system. In the example, this is LINK1, the logical link name specified in the originating system on the MSNAME macro at system definition. However, if the message is subsequently sent back to the originating system, the originating LTERM name is reinstated in the first field of the IOPCB.

- IMS turns on a bit in the field of the IOPCB that is reserved for IMS. This is the second bit in the first byte of the 2-byte field. When this bit is on, it indicates that this message is associated with directed routing and the LTERM name field of the IOPCB contains the MSNAME.
- If ROUTING=NO is specified on the TRANSACT macro, there is no change in the contents of the IOPCB. The default is NO.

The Link Receive Routing exit (DFSCMLR0) routine in the processing IMS system can be used to determine the destination transaction code or LTERM name if directed routing was used to send the message.

Restrictions on the use of directed routing include:

- The multiple systems verification utility cannot detect errors associated with MSC directed routing.
- The /MSVERIFY command cannot detect errors caused by the improper use of MSC directed routing.
- Directed routing does not support a program-to-program switch from a non-conversational transaction to a conversational transaction. For example, a non-conversational transaction in system A could not use directed routing to perform a program-to-program switch to invoke a conversational transaction in system B.
- Response mode cannot be propagated on a DL/I ISRT call in a directed-routing transaction.
- To maintain system integrity and prevent errors, an IMS system in a multisystem configuration verifies all specified destinations, unless MSC directed routing is used. When MSC directed routing is used, IMS ensures only that a program-to-program switch is not being done from a non-conversational transaction to a conversational transaction.
- Transactions received by directed routing on a link are passed to the transaction authorization module for authorization checking. However, the password is not passed across the link, so transaction authorization checking fails if a password is required. Transactions not requiring a password can be accepted.

Although these restrictions are not trivial, directed routing does provide an application-layer level of routing control within an IMS sysplex using MSC.

Chapter 6. Identifying and Solving MSC Problems

MSC can play a very large part in an IMS/ESA sysplex environment, so it is vital that you obtain complete documentation when problems occur and effectively analyze the material gathered. In this chapter we explain how to obtain the type of documentation required for MSC connection problems and show what a traced successful message send-and-receive flow looks like.

6.1 MSC Communication Task Trace

The main tool used for MSC problem identification is the MSC communication task trace.

The trace can map the data and process flow for MSC link traffic during restart, normal and abnormal message flow, and shutdown on the basis of when the trace is turned on. The output produced for a single transaction can be quite large, so activate the trace as close to the instance of the problem as possible and turn it off immediately after the problem is encountered.

Enter this command on both link partners to initiate the trace for a specific link:

/TRACE SET ON LINK X LEVEL 4 MODULE ALL

The ALL parameter indicates that device dependent modules (DDMs), the MSC communications analyzer, and message format service (MFS) module interfaces will be traced. Specifying *LEVEL 4* engages all related communication and MSC control-block structures, queue manager and I/O line buffers, and IMS save-area sets used in following the module processing flow.

To turn tracing off, enter the following command:

/TRACE SET OFF LINK X

Trace entries can be selected and formatted by an IMS/ESA-supplied program, DFSERA10, referencing formatting routine DFSERA30.

Sample JCL for DFSERA10 to Select and Print MSC Trace Records

```
//STEP01    EXEC  PGM=DFSERA10
//STEPLIB   DD  DSN=IMS51.RESLIB,DISP=SHR
//SYSUT1    DD  DSN=MSC.SAMPLE.SLDS,DISP=SHR
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
CONTROL  CNTL
OPTION   PRINT O=5,T=X,V=6701,L=2,EXITR=DFSERA30,C=E
END
```

Log records with an identification of X′6701′ are selected and formatted in this manner. This is similar to the process used to trace and format IMS line and node activity.

6.1.1 Communication Analyzer Trace Points

The MSC communications analyzer controls the necessary logical functions to cause message editing, queuing, reading, writing, and routing. It has internal routines that are traced on entry with specific trace codes associated with them as shown in Table 4

Table 4. MSC Communications Analyzer Entry and Trace Points

Entry Point	Trace ID	Function
1	AM01	Process input from a link
2	AM02	Read the link
3	AM03	Determine what to do next on the link
5	AM05	Write to the link
6	AM06	Perform end of write function
8	AM08	Return a message to the message queues for later transmission
9	AM09	Generate an error message
10	AM10	Quiesce the link
12	AM12	Wait for asynchronous I/O or the enqueue of a message

6.1.2 Device Dependent Module Trace Points

The MSC communications analyzer interfaces with specific DDMs, which, by interfacing with other functions, perform the I/O link control functions specific to the link type. The DDMs also have entry points with associated trace codes as shown in Table 5.

Table 5. MSC Device Dependent Module Trace Points

Entry Point	Trace ID	Function
1	DM01	Set up for write operation
2	DM02	Check for errors after write
3	DM03	Set up for read operation
4	DM04	Check for errors after read
7	DM07	Connect or disconnect the link

6.2 MSC Control Block Structures and Message Routing

The trace review that follows refers to control blocks. These structures and the part they play in the MSC process flow are explained.

If we examine the major control block structures associated with a horizontally partitioned configuration using a single network front-end IMS system linked through MSC to back-end IMS application processors, we see that one system has all of the terminal definitions and the other systems only transactions and remote terminals. Table 6 on page 43 shows the control block structures associated with this environment. We have included the names of the mapping macros if you want to assemble the structures to obtain more information.

Table 6. Control Block Structures Associated with the MSC Environment

Control Block	Description	Mapping Macro
Terminal Related		
CLB Communications line block	One per VTAM node	ICLI CLBBASE=0
CTB Communications terminal block	One per PTERM and subpool	ICLI CTBBASE=0
CNT Communications name table	One per LTERM	ICLI CNTBASE=0
CRB Communications restart block	Involved with MSC link synchronization	ICLI CRBBASE=0
RCNT Remote CNT	If the target of a CHNG call is an RCNT, the message is enqueued on the logical link to the IMS system where the physical terminal actually exists.	ICLI CNTBASE=0
Application-Related		
SMB Scheduler message block	One per transaction	IAPS
RSMB Remote SMB	One per remote transaction (mapped by SMB structure). Has remote and local SYSID numbers in it.	IAPS
Transaction class table (TCT)	One per IMS system to queue SMBs off for the scheduling process	DFSTAB
MSC-Related		
LCB Link control block	Represents the link	LCB
LLB logical link block	Mapped by CLB for the link	ICLI CLBBASE=0
LNB logical link name block	Mapped by CNT	ICLI CNTBASE=0
LXB Link extension block	Event control block (ECB) or block for asynchronous access method activity.	LXB
SYSID table	Table of defined SYSIDs	

6.2.1 Association with Event Control Blocks and IMS ITASKS

To perform work directly associated with link structures, the LLB has in its prefix an event control block (ECB) that is the key to dispatchability in MVS environments. When the ECB associated with this LLB is posted, the MSC communications analyzer DFSCMS00 receives control and begins servicing the link requests. As with task control blocks (TCBs) in MVS, the ITASK is the fundamental dispatchable unit of work in IMS. Posting an ECB expands to IPOSTING an IMS ITASK.

The prefix of the link extension block (LXB) also contains an ECB. During initialization, each link has an LXB assigned to it, and all reads and writes to a link are issued by specifying the LXB′s ECB. Thus a certain amount of parallel processing between link I/O and other link functions can occur.

6.2.2 Message Routing

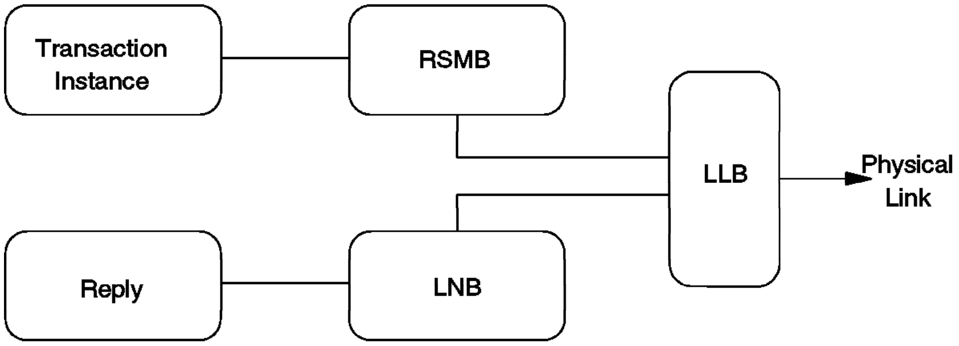
Each message involved in the MSC environment has information contained in prefix-segment items that are attached to the front of the message before it is sent over the link. This prefix contains information necessary to route the message to the proper system and ensure that responses are returned to the input terminal. The information contained in the message prefix includes:

- LTERM name
- User ID, if present

- Destination name
- Destination SYSID
- Origination SYSID
- SYSID trace

The prefix data is not available to the application program.

When IMSA receives a transaction code from a terminal, the message destination—in this case, a schedules message block (SMB)—is determined, and the transaction is enqueued. If the SMB is defined as remote, the message router determines the link (represented by an LLB) to which the transaction is assigned and queues the remote SMB (RSMB) off the LLB by priority. Figure 10 shows the MSC control blocks involved in this scheduling process.



All transactions enqueued on SMB's. If they are destined for a remote system, the SMB is called an RSMB (Remote Scheduler Message Block)

The RSMB's are queued off the LLB (Logical Link Block) describing the physical link.

All responses are enqueued from the LNB (Logical Link Name Block)

Figure 10. MSC Scheduling Control Blocks

The LLB is then posted and the LLB ITASK is given control to retrieve the message from the queue and send it to IMSB. This I/O occurs under control of the LXB ITASK.

When IMSB receives the message, it stores the source LNB name in the CTB for use by the queue manager when an IMS reply is sent back to IMSA. The source LNB represents the linkage back to the terminal originally entering the transaction on IMSA. When the reply is inserted into the message queue, the queue manager enqueues the message to the LNB that was saved. The message router then determines to which link this LNB is assigned and IPOSTs the applicable LLB.

The destination and origination SYSIDs in the message prefix are the primary routing fields. When a message is received at a remote IMS system, the SYSID is checked to see whether this is the destination. If not, the SYSID table indicates over which link the message must be sent to reach its ultimate destination.

6.3 MSC Transaction Routing and Response Flow

Table 7 describes the flow of messages between two link partners. Local system IMSA receives the input message and a transaction is scheduled. It performs a program-to-program switch to a remote transaction. The input is received on the remote site, and a response is sent back to the input terminal. The event column lists the normal log records. (Events 01, 03, 31, 35 and records such as AM03 and DM01 reflect the fact that this activity was traced to map the MSC communications analyzer and MSC DDM flow of control.)

Table 7 (Page 1 of 4). Successful Transaction Routing and Reply Flow

Event	IMSA	IMSB
01	Place input message from terminal into MSGQ buffer	
35	Message is enqueued to its destination. The SMB is queued from the TCT by class as part of the scheduling process.	
08	Application is scheduled.	
5607	Start of unit of recovery	
31	GU to message queue for input record	
03	Message put into queue buffer	
35	Message is enqueued. At this point the LLB that has an ECB in its prefix is posted, causing the MSC communications analyzer to be dispatched.	
3770	Syncpoint record	
3710	Syncpoint record	
33	Queue manager (QMGR) releases the input record.	
5612	Phase 2 commit processing ends.	
07	Application terminates.	
AM03	What′s next. Check whether any output is available; in this case, yes.	
31	GU call from MSC to QMGR for output. Validate the SYSID at this point. At this point the logical link is in a busy state. Access to the queued data may require I/O to a message queue data set.	
DM01	Write setup. Check to see what to send, set up output buffer, and move message segment from queue buffer (QBUF) to output buffer.	
CM02	Issue QMGR Get Next (GN) output message.	
66	MSC pre-dequeue recovery record logged through CHKW because this transaction is recoverable. At this point, a flag is set (CTB8DAT bit in CTB) indicating that a response is required from IMSB for the message to be shipped. No more data can be sent on this logical link until the response has been received and processed.	
DM0I	Entry to access method. An EXECRPL macro is used for the SEND request to VTAM. The setting of Pacing or a low class of service (COS) value and delays in the physical transmission might elongate transmission times.	

Table 7 (Page 2 of 4). Successful Transaction Routing and Reply Flow

Event	**IMSA**	**IMSB**
AM05	Do write. Then return to IMS dispatcher to wait for read or write interrupts.	
DM02	Write interrupt. Still have not received the response from IMSB to the last message shipped so no more can be sent at this time. VTAM return codes are checked. Also bit CRB2SYNC in CRB communications restart block (CRB) set to indicate ″in sync interval″ for recovery purposes. It will be reset when the definite response (DR2) from IMSB is received.	
AM03	What′s next. Are there any commands or responses to be sent?	
DM01	Write setup. IMSB owes IMSA a definite response for the data request unit (RU) because bit CTB8DAT is still on, so IMSA activity for this link goes idle waiting for a response.	
AM12	Idle link. Wait for a response from IMSB.	
DM04		Read interrupt. Flag set indicating that IMSB owes IMSA a response for the message just received. No messages can be sent on this logical link until the response is sent. Also, return codes and message sequence numbers are checked along with validation of the data block length in the receive buffer.
66		MSC pre-enqueue recovery record
AM01		Process input. The SYSID is obtained and checked for local or remote associations (it is local). The message destination is located, and security checking occurs if requested.
01		Received input message placed into IMSB (QBUF) by a call to the QMGR. Increment the message received count for this link. Possible message queue data set I/O could occur at this point.
35		Message enqueue on permanent destination. The SMB is enqueued on the TCT for the scheduling process.
08		Application scheduled in IMSB
5607		Start of Unit of Recovery
31		GU to MSG queue by a call to the QMGR
DM03		Read setup to reissue the Receive
DM0I		Entry to access method. The VTAM RECEIVE macro is issued now.
AM03		What′s next. Any messages or responses queued to be sent to IMSA? Yes, there are.
DM01		Write setup for response to IMSA
DM0I		Entry to access method to send the response
AM05		Do Write: 0 length RU (DR2) ACK response.

Table 7 (Page 3 of 4). Successful Transaction Routing and Reply Flow

Event	**IMSA**	**IMSB**
DM04	Read interrupt for response from IMSB. The response-owed flag (CTB8DAT) is reset, and control is passed to the analyzer. The logical link is now out of Busy status.	
AM06	After good write, increment the message sent count.	
DM02		Write interrupt. Check the return codes.
36	Call the QMGR to dequeue the message originally shipped to IMSB. Reset the CRB2SYNC ″in sync interval″ bit.	
33	QMGR release of message. Update the committed output sequence number. Dequeue the RSMB representing the transaction off the LLB.	
AM03	What′s next. Prepare to issue re-receive.	
AM03		What′s next
DM03	Read setup. Set up request parameter list (RPL) for receive.	
DM01		Write setup
AM12		Link idle
DM0I	Entry to access method. Execute EXECRPL to issue the receive.	
AM03	What′s next. Anything else to send?	
DM01	Write setup. Nothing to send. Prepare to go into Idle mode.	
AM12	Set link idle and Itask inactive.	
03		Message in QBUF. Application has inserted reply back to IOPCB.
35		Message enqueue from the input LNB. A Type 35 log record is committed through a CHKW call.
3770		A syncpoint record is logged through a WTWT call.
3710		A syncpoint record is logged through a WTWT call.
5612		Phase 2 processing ends.
07		Application termination
AM03		What′s next
DM01		Write setup for response back to input terminal in IMSA
CM02		GN
66		An MSC recovery record is logged to the write-ahead data set (WADS) through a CHKW call.
DM0I		Entry to access method
AM05		Do write of application response to be shipped to the input terminal on IMSA. A VTAM SEND is issued.
DM04	Read interrupt after good write	
66	An MSC recovery record is logged through a CHKW call.	

Table 7 (Page 4 of 4). Successful Transaction Routing and Reply Flow

Event	IMSA	IMSB
AM01	Process input. At this point the reply message from IMSB is ready to be sent to the input terminal.	
DM02		Write interrupt
AM03		What′s next
DM03	Read setup	
DM0I	Entry to access method	
AM03	What′s next	
DM01	Write setup	
DM01		Write setup
AM12		Link idle
DM0I	Entry to access method	
AM05	Do write: 0 length RU (DR2) ACK response.	
DM04		Read interrupt
AM06		After good write
DM02	Write interrupt	
AM03		What′s next
AM03	What′s next	
DM03		Read setup
DM0I		Entry to access method
AM03		What′s next
DM01	Write setup	
DM01		Write setup
DM0I		Entry to access method
AM12	Link idle	
AM12		Link idle
36		Output message is dequeued.
33		QMGR release of output message. The logical link is now available for more message flow activity.

The amount of traced processing for a simple transaction routing and response is extensive, but the key points from this example are:

- Use of the MSC control block to interface with the normal communication and application scheduling structures.
- Read and write activity to the message queues that might involve I/O to the message queue data sets.
- Forced check writes of important log records and message sequence checking to ensure synchronization between the two sides of the link.
- Serialized message shipment and expected zero length record response before any more link activity can occur.
- VTAM pacing, COS values, Network Control Program (NCP) delays, and physical transmission times have a bearing on MSC-VTAM traffic flow rates.

6.4 Conversational Processing with MSC

Now that we understand MSC message flow and how some of the IMS control block structures are used, we can expand the scope of discussion to include conversation transaction processing in an MSC environment.

Conversational transactions have to be defined identically in the local and remote systems, with the SPA being fixed and of equal length for all programs involved in the conversation.

The conversational characteristics of the transaction, such as the SPA and conversation control block (CCB), are controlled by the input system for the duration of the conversation.

Let us follow the flow of a conversational transaction in an MSC environment.

1. The input system receives a message and passes the first segment to the Terminal Routing exit, as is done for messages before the type and destination have been determined.
2. The message is recognized as the start of a conversational transaction to be processed by a remote system. The SPA is inserted as the first segment, and the message is routed by MSC to its destination.
3. The remote system checks the attributes of the transaction against the local definitions. If this check fails, the SPA is returned to the input system and the conversation is terminated. If the transaction is accepted, it is added to the message queue.
4. The conversation is processed, and the response, together with the SPA, is returned to the input system, where the inserted SPA is checked for validity.
5. If the response is destined for a local terminal, the CNT is verified to be connected to the same CTB as during input.
6. If the conversation is to continue, the transaction code is located and placed in the CCB. The SPA is updated and added to the message during the next conversational step. MSC routes both the SPA and message to the SMB, saved earlier in the CCB.

The terminal routing exit will not be invoked for the continuation of the conversation.

All abnormal termination processing for a conversational transaction is handled by the input system. If the transaction abends in the remote system, the SPA is returned to the input system, and the transaction code and program are stopped on the remote side.

For all messages causing the start of a conversation under the stopped transaction code, the remote system returns a warning message (DFS065) to the input terminal.

If an active conversation produces further messages for a stopped transaction, the messages are queued at the remote side. No warning message is presented to the terminal operator, but one is shipped to the master terminal.

The /EXIT command for conversations being processed reacts quite differently in a multisystems environment than in a single system environment:

- The /EXIT command can be entered while a conversation step is executing in a remote system. It also can be entered when the transaction to, or the response from, the remote system is being shipped over the link.
- The Conversation Abnormal Termination exit, DFSCONE0, is called with the most current SPA for the conversation and an indicator that the final SPA will be presented later.
- The application processes and completes its work in the remote system, and, after the response arrives, DFSCONE0 is invoked a second time and should be able to complete processing to clean up the conversation.

6.5 Matching MSC Traffic Flows between Partner Systems

There is an inherent difficulty in matching sent and received messages when shipped across multiple IMS systems linked by MSC. In the above trace example, we have only one message containing the routed transaction and resultant response; in normal production environments, however, many messages flow between MSC-attached IMS systems.

No direct relationship exists between the relative record number in a device queue for a message in one partner system and that in the other partner system. This is true for both long and short message queue data sets. As a result, the logged output and input segments cannot be aligned by record number. The method we used was associated with the time stamps contained within the prefix of the input and output records as logged through the partner systems.

For example, let the local system IMSA route a message across an MSC link to remote system IMSB where the transaction will be executed. The prefix of the type 03 output record on the IMSA log and the type 01 input record on the IMSB log contains three time-stamp fields. Type 01 and 03 records use the same log record mapping. Assemble the QLOGMSGP DSECT for more detail.

The first time-stamp field is held at label MSGETIME within the message prefix system segment. MSGETIME is set just before the type 35 enqueue log record is created. It is the time of the message enqueue in the IMSA sending system and is available on the type 03 record from IMSA and the type 01 record from IMSB.

The second time-stamp field is mapped using field MSGMSCTM within the MSC system segment. This is the time of the enqueue of the message in the receiving system IMSB and is available in the type 01 input log record.

The format of these two fields is HHMMSSTF, so value 1934081F equals a 19:34:08.1 time stamp. The minimum timer value of 0.1 second is not sufficient to differentiate messages on a busy link.

The last time-stamp field in the message prefix does have the required granularity of timer units. In the message system extension within the type 01 and 03 message prefixes, field MSGSSCLK maps an 8-byte system clock value:

- In the type 03 log record from IMSA, this store clock value is associated with the original enqueue of the message onto the message queue in IMSA.
- This also provides the enqueue time on the receiving IMSB system in the type 01 log record.

For example, if the type 03 output record logged by IMSA has:

MSGSSCLK value = AC960969 DCAF2207 a conversion routine calculates the date and time stamp as

Date = Mar 18 1996 , Time = 19:34:08.073458 The type 01 input record logged by IMSB has:

MSGSSCLK value = AC960969 E4DFA606 which results in:

Date = Mar 18 1996 , Time = 19:34:08.107002

This provides the necessary accuracy to match specific MSC link transfers.

Note: IMS/ESA does not have a conversion routine available, so we used a local program for these calculations.

6.6 Other MSC Diagnostic Log Record and Trace Aids

Other facilities assist in the problem identification process in MSC environments:

- **MSS1 and MSS2 Records**

  If an I/O error (correctable or not) occurs, all available control blocks are SNAPed regardless of whether or not the link is being traced at the time. MSS1 and MSS2 records have the same format as type X′6701′ log records and can be selected and formatted by utility DFSERA10 with formatting routine DFSERA30. A type 03 output log, containing the message shipped to the master terminal as a result of the error, follows the MSS1 and MSS2 records.

- **MSVID flow trace field**

  Field MSGMSVID within the MSC system segment item in the prefix of the type 01 and 03 log records is a double word trace of MSC IDs for the last eight IMS systems through which messages were routed. It is initialized when a terminal sends a message or when an application program does an ISRT of a message and is updated for each intermediate system and the destination system. The low-order byte in the trace contains the most recent entry, and each additional entry results in a shift left. The MSC ID comes from the MSVID parameter, which is the value specified in each system′s IMSCTRL macro during system definition.

  This particular trace will be of use in sysplex environments where several local, intermediate, and remote IMS systems are connected together through MSC links.

For more information, please see Chapter 10 (MSC Service Aids) in *IMS/ESA V5 Diagnosis Guide and Reference*, LY27-9620-00 (available to IBM licenced customers only).

6.7 Obtaining Dumps for MSC Problems

In a sysplex environment, gathering all of the information necessary to ensure that a problem can be solved is highly complex. The data required to investigate errors is potentially spread across a large number of address spaces.

Below we list some general recommendations for setting up IMS and MVS to obtain as much valuable information as possible when the situation requires dumping one or more MSC partners in a sysplex. This could occur when one of the partners appears to be looping or to be hung during link restart, transmission, or shutdown or when the system has forced an SVC dump because of nonrecoverable failures.

- If an IMS control region seems to be hung or in a loop and a dump is in order, dump the XCF, CONTROL, DL/I, IRLM, and DBRC address spaces using the SDUMP MVS command.
- Specify the dump options of SDATA=(XESDATA,ALLNUC,CSA,LSQA,RGN,PSA, SQA,SUM,SWA,TRT) in the SYS1.PARMLIB(IEADMR00) member.

  Dump commands to retrieve all required data can be long, and it is easy to enter them incorrectly. Here is a process that is a lot less error-prone: set up a series of IEASLPxx members in SYS1.PARMLIB that can be activated, using the SET SLIP MVS command, when a dump is required.

  The following are the recommended settings for the relevant products:

  Member for IMS Dumping in SYS1.PARMLIB(IEASLPIM):

  ```
  SL SET,IF,(N=IEAVEDS0,00,F),ID=IMSDUMP,
      JOBLIST=(XCF*,IRLM*,DLI*,DBR*,IMS*),
      SDATA=(RGN,XESDATA,ALLNUC,CSA,LSQA,PSA,SQA,SUM,SWA,TRT),
      REMOTE=(JOBLIST=(XCF*,IRLM*,DLI*,DBR*,IMS*),
      SDATA=(RGN,XESDATA,ALLNUC,CSA,LSQA,PSA,SQA,SUM,SWA,TRT))
  ```

  IRLM*, DLI*, DBR*, and IMS* represent a generalization of the names of the IRLM, DLISAS, DBRC, and IMS control region address spaces. These names differ in different installations.

  Member for IRLM Dumping in SYS1.PARMLIB(IEASLPIR):

  ```
  SL SET,IF,(N=IEAVEDS0,00,F),ID=IRLM,
      JOBLIST=(XCF*,IRLM*),
      SDATA=(RGN,XESDATA,ALLNUC,CSA,LSQA,PSA,SQA,SUM,SWA,TRT),
      REMOTE=(JOBLIST=(XCF*,IRLM*),
      SDATA=(RGN,XESDATA,ALLNUC,CSA,LSQA,PSA,SQA,SUM,SWA,TRT))
  ```

  In the above SLIP SETs, IEAVEDS0 is the MVS dispatcher. The SLIP for an ID of IMSDUMP or IRLM will be matched very shortly after the SET SLIP= command is issued. Since MVS must have been responding for the SET command to be issued, this location within the MVS dispatcher responds as though you were issuing the same command on the console expecting an immediate dump command.

- Ensure that the SYS1.DUMPxx data sets are large enough to contain up to four IMS regions and IRLM in each. The regions include CTL, DLI, DBRC, and possibly one dependent region. In large installations, the required size can be more than 500 cylinders of 3390 DASD. MVS V5.1 features dynamic allocation for dump data sets, so you do not have to allocate the dump data set beforehand.
- Specify IMS control region EXEC parameter value of FMTO=D. This produces an SDUMP for terminating and nonterminating errors. SYSMDUMP, SYSABEND, or SYSUDUMP are produced only if SDUMP fails.
- To determine what is dumped by a SVCDUMP, under IPCS on the command line issue:

  IP STATUS WORKSHEET

  Go to the bottom of the output and scroll up to find SDUMP PARMLIST and SDATA. These provide the current options.

Or, at the console, type

D D,O

Scroll down to SDUMP to find the specified options.

- Place a SYSMDUMP DD statement in the JCL of the IMS Control, DL/I, and DBRC regions, making each SYSMDUMP data set large enough to contain a dump of the control region to ensure you have enough space.
- Place a SYSUDUMP in the JCL of the IMS dependent regions by placing SDATA=(CB,ERR,SUM) and PDATA=(JPA,LPA,PSW,REGS,SA,SPLS) in SYS1.PARMLIB(IEADMP00). If the online formatting for large dependent regions takes too long, use SYSMDUMP. However, be aware that SYSMDUMPs can concatenate to one another.
- Set the IMS Dispatch, Scheduler, DLI, and Lock traces on using the IMS PROCLIB member DFSVSMxx with DISP=ON, SCHD=ON, DL/I=ON, LOCK=ON; or issue the /TRACE SET ON TABLE nnnn command where nnnn= DISP, SCHD, DL/I, and LOCK. The maximum performance degradation per trace is 1% to 1.5%, although many customers find the degradation too small to measure.
- Set the MVS system trace table size to 999 KB by placing the TRA ST,999K command in the MVS COMMNDxx PARMLIB member. Note that dumps specifying TRT as SDATA options increase in size.
- Set the MVS master trace table to 100 KB by including TRA MT,100K in the SCHEDxx member of SYS1.PARMLIB.
- Ensure that the IMS offline dump formatter (ODF) is installed. For more information, review the following manuals:

  IMS/ESA V5 Diagnosis Guide and Reference, LY27-9620-00 (available to IBM licenced customers only)

  IMS/ESA V5 Utilities Reference: System, SC26-8035-00

Although the above recommendations imply some level of trade-off between storage and processing cycles for first time failure data capture, given the complexity associated with the diagnosis of MSC problems in sysplex environments, the investment is worthwhile.

Chapter 7. MSC Performance

Throughput associated with the routing of transactions and messages through a sysplex environment using MSC is a very important element of sysplex-wide performance. In this chapter we discuss MSC performance considerations, including measurement tools, and present the results of performance benchmarks run at the IBM Santa Teresa Laboratory in San Jose, California.

Many factors in a sysplex may influence performance in a MSC environment. We review transaction specifications, buffer settings and system data set placement, the use of parallel sessions, and general VTAM considerations.

7.1 Transaction Specifications

In this section, we describe the parameters affecting the performance of transactions that have been transferred between IMS systems via MSC.

7.1.1 Link Priorities

Transactions specified as remote must be defined in both the input system and the destination system. However, the priority specifications on the TRANSACT macro may differ on the two systems. The TRANSACT macro priority specification is set in the PRTY parameter:

```
TRANSACT PRTY=(NORMAL, LIMIT, LIMIT COUNT)
```

NORMAL

The priority assigned to this transaction when the number of input transactions queued and waiting to be processed is less than the limit count value. The valid specification range is from 0 through 14. The default is 1.

LIMIT

The priority to which this transaction is raised when the number of input transactions queued and waiting to be processed is equal to or greater than the limit count value. The valid specification range is from 0 through 14. The default is 1.

LIMIT COUNT

The number that, when compared to the number of input transactions queued and waiting to be processed, determines whether the normal or limit priority value is assigned to this transaction. The limit count value can range from 1 through 65535.

The priority of a transaction defined as remote becomes the priority at which it is eligible for processing across the link. Therefore, even if the transaction is given a high priority in the input system, processing and a possible response may be delayed because of a low priority on the destination system. Even when the priorities are not different, it is important to understand how transactions and responses are queued for processing on the link. Figure 10 on page 44 shows the MSC control blocks involved in this scheduling process.

7.1.1.1 Link Priority Enqueue

The LLB represents the link, so all traffic for the link is enqueued from this control block. All remote transactions are first enqueued on a first-in-first-out (FIFO) basis on the corresponding RSMB. The RSMB is then enqueued from the LLB at the current priority of the transaction.

All responses (inserts to the IOPCB or alternate PCB) are associated with the MSNAME as mapped through the LNB. The responses are enqueued on the proper subqueue, according to the current priority of the transaction that inserted them to the message queue. The LNB is queued from the LLB at priority 7, and responses are queued from the LNB on four subqueues:

Q1 = Priority 12—15

Q2 = Priority 8—11

Q3 = Priority 4—7

Q4 = Priority 0—3

7.1.1.2 Link Queuing Example

Figure 11 presents a transaction and response queuing example.

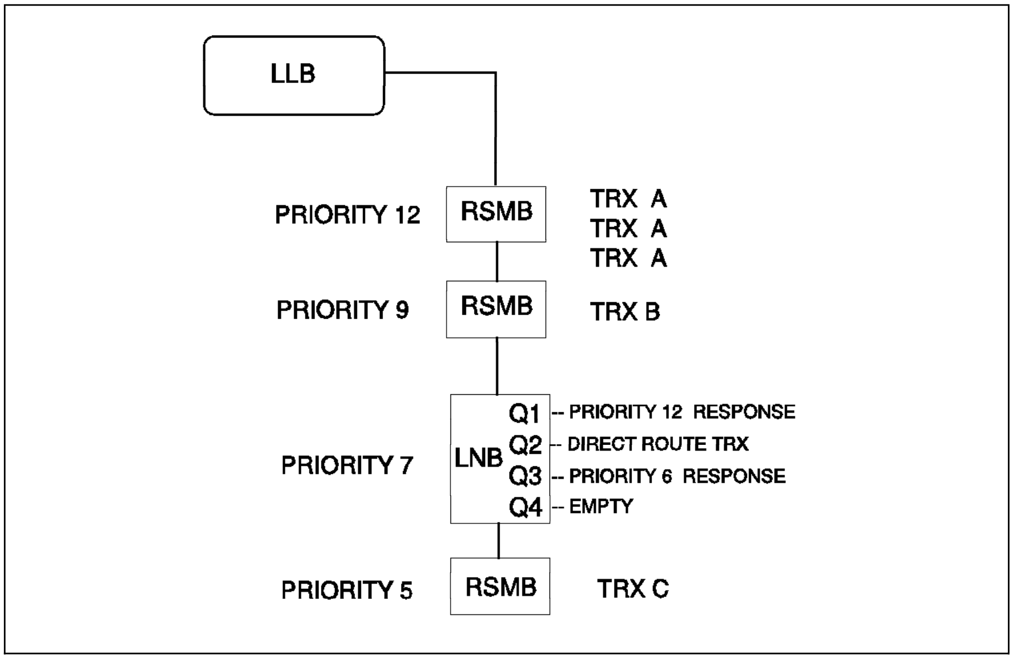


Figure 11. Transaction and Response Queuing Example

Three messages for transaction A are queued from the RSMB, representing a remote application. The RSMB is queued from the LLB at priority 12; the TRANSACT macro PRTY value is 12 either as its normal setting (the limit count had been reached) or because an operator command altered it to priority 12. The LLB-RSMB structure represents a portion of the necessary linkage for input scheduling processing.

After the transaction is sent over the link, it is put in the message queue of the destination system and scheduled according to the local transaction priority. This priority may differ from the value in the originating system.

The LNB represents the response destination linkage at this point and any response message is queued from it. A response queued on the LNB is set at the priority of the processing transaction.

The LNB is queued on the LLB at priority 7. Only after all transactions with a priority higher than 7 are transmitted will processing of responses from the LNB be scheduled.

In our example, the response traffic queued from the LNB will be processed as follows. The priority 12 response is shipped back first, followed by the directed routing transaction at a priority that sets it onto the Q2 (priority 8—11 range), and finally, the priority 6 response is queued onto Q3 in the LNB.

Therefore, consider the scheduling algorithm when defining remote transactions, because:

- Any transaction with an initial priority lower than 7 is eligible for link transmission only after all transaction responses have been sent. This is the case for transaction C in our example.
- Specify printer transactions with a priority of 1 to 3 and place them in LNB Q4, so they do not interfere with other responses. If your sysplex is one that creates large printer reports, the response time of low-priority transactions may be affected.
- Check NORMAL and LIMIT priority settings carefully. If they fall into two different LNB subqueues, out-of-sequence response situations might occur.

7.1.1.3 Inquiry-Only Transactions

The TRANSACT macro also contains a specification of whether the transaction is an inquiry transaction:

```
TRANSACT INQUIRY=(NO]YES, RECOVER]NORECOV)
```

If the transaction is defined as inquiry, it may also be defined as nonrecoverable. If nonrecoverable, the number of check writes (CHKWs) to the log for the transaction is reduced from 2 to 1 on each side of the link. This reduction aids in link performance.

7.2 Buffers and System Data Sets

In this section we describe the buffers that MSC uses and the system data sets that must be defined.

7.2.1 MSC Buffers

The size of the buffer used to send a message to a link partner is specified by the BUFSIZE keyword in the MSPLINK macro. Define this buffer large enough to hold the longest message segment or SPA for conversational processing.

If the buffer size is too small, the output data is split into two or more parts, and each part must be handled by a separate VTAM SEND/RECEIVE. MSC attempts to put as many segments of a multisegment message into the MSC buffer as will fit.

Defining buffers larger than the message or SPA size does not improve performance, because MSC does not attempt to place more than one output message in a buffer.

7.2.2 Message Queue LRECL

MSC adds an MSC prefix to all input and remote output messages. It also adds a 40 byte MSC extension item to all messages if MSC is generated at the IMS/ESA V5.1 level. The basic, system, system extension, and workload manager prefix segments are not sent across MSC links, although all other prefix segment items are. The prefix segment items not shipped between systems are rebuilt on the receiving IMS when necessary. If the LRECL of the short message queue set data set is defined exactly to distribute messages between the short and long message queues, an additional prefix should be considered.

When using MSC to connect IMS systems of different releases, it is important to take into consideration all the message types (such as ISC, APPC, and OTMA) and the prefix sizes that accompany them. It is recommended that the message queue LRECL and block sizes be identical across IMS systems. If this is not possible, the LRECL and block sizes should at least be large enough to hold the message prefix and segments without spanning across QPOOL buffers.

7.2.3 Message Queue Data Set I/O

Message queue I/O may cause severe performance degradation in an MSC environment, because the I/O causes delays for the entire logical link, not just for a specific user or transaction. If one of the link partners is unable to receive messages from the other system because of a /DBRECOVERY command or a checkpoint being processed, the originating system has to keep messages intended for transmission in its message queues. If the message queue pool is not large enough, or the time of the link suspension is too long, messages are flushed to the message queue data sets on a least-recently-used basis.

When the link partner is again ready for transmission, the messages to be sent first have to be read in from the message queue data sets, suffering I/O delays in the process. Therefore, it is necessary to carefully tune the size of the message queue buffer allocations according to workload and expected production delays throughout the sysplex. The MSC buffer size specified on the MSPLINK macro in the BUFSIZE= parameter should be large enough to hold the largest message segment or SPA, including the message prefix.

The use of multiple message queue data sets should also be considered to allow for parallel access to the message queue data sets in the event that messages are flushed to DASD. Placement of these data sets on separate channel paths is recommended for efficiency of access.

7.2.4 Write Ahead and Online Log Data Sets

The response time of the write ahead data set (WADS) is also critical to MSC performance because of the additional CHKW calls issued. Placement of the WADS and OLDS is more critical in an MSC environment because the I/O to these data sets influences the time a message transmission takes, as the message has to be logged before it is eligible for transmission. Place the WADS on devices with as little interference as possible. A 2 to 3 ms device service rate is near the best average that can be expected.

If available, use DASD fast write (DFW), which performs write operations at cache speeds. The WADS gains performance benefit from DFW. In a typically busy IMS system, the WADS tracks remain resident in the cache and benefit from DFW.

But the trend today is to define many OLDS buffers, each backed by its "own" WADS track group. If a WADS track group is available for each OLDS buffer and a large number of OLDS buffers are specified, DFW may not produce cache hits because the WADS track groups are reused in a round-robin fashion, causing them to be flushed out of the cache before they are written to.

7.3 Use of Parallel Sessions

The initiation of parallel sessions for one physical link helps to alleviate queues building up at the link for transmission under normal circumstances. In most cases, it is not the capacity of the physical link that is a problem but the utilization of the logical link (the time the LLB is busy or waiting for an acknowledgment). Parallel sessions are available only when VTAM is used as the MSC communications access method.

The parallel session specification is set by:

MSPLINK SESSIONS=X

where X=1 to 255.

The logical link is used while the message is:

- Logged in the local system
- Transferred to the remote system
- Queued and logged in the remote system

The logical link becomes available only after an acknowledgment is returned to the local system.

The IMS startup parameter, NLXB, is used to increase the SESSION value specified at system generation time. NLXB specifies the number of parallel sessions to be added during IMS control region startup. The NLXB value is added to the value specified in the SESSION parameter to increase the number of LXBs generated for each link. The default for NLXB is 0; NLXB can have a maximum value of 255. Each defined parallel session uses approximately 50 bytes of fixed common storage area (CSA).

In the benchmark discussions in 7.6, "MSC Performance Benchmark" on page 62, we present the effect of increasing the number of logical links on MSC message traffic throughput. MSC does not offer a balancing algorithm for parallel sessions, so either user exits or the workload router (described in Chapter 8, "IMS/ESA Workload Router" on page 69) must be used.

7.4 VTAM and Physical Link Considerations

For those installations using VTAM as the access method with MSC, a few areas of performance are worth reviewing:

- **VTAM transmission priority**

  The transmission priority is specified in a COS table, which is pointed to by the VTAM mode table. The higher transmission priority of 2 causes VTAM to initiate data transfer as soon as a message arrives from IMS. A Lower transmission priority may cause delays in transfer.

- **VTAM IOBUF size**

  Together with MAXBFRU, the IOBUF size determines how many buffers VTAM has to chain to handle a particular RU.

- **VTAM MAXBFRU specification**

  MAXBFRU specifies the amount of storage VTAM uses to receive the data from the channel-attached resource. The value is the number of 4KB buffer pages. The buffer size must be large enough to contain the largest anticipated data transfer.

- **VTAM protocol**

  Because MSC uses RECEIVE SPECIFIC calls rather than RECEIVE ANY, the number of IMS RECANY buffers specified at IMS/ESA startup has no impact on MSC performance.

- **Pacing**

  Do not set Pacing on for MSC physical links.

The NCP delay should be set to 100ms when MSC is part of the network environment.

7.5 MSC Performance Measurement Tools

To monitor MSC performance, data from several sources is usually necessary. In this section we discuss some of the major sources of performance data that can be used.

7.5.1 IMS DC Monitor

Although there are no specific timings for MSC events, MSC reports can indicate whether the delays arise in a specific link to a specific system or are related to more global factors.

Three IMS monitor reports highlight message events associated with MSC:

- **MSC Traffic Report**

  Presents the enqueue and dequeue counts of messages that use the various link paths for the Monitor interval.

- **MSC Summary Report**

  Shows summaries of traffic queues for each:

  - Input transaction name
  - Destination name

  - Link number
  - Destination system

  The effects of link loading can be assessed through this report.

- **MSC Queuing Summary Report**

  When the local system is an intermediate system, this report is generated to show intersystem messages queued on the intermediate system before being sent to the destination system. Link queuing times can be assessed using this report.

You can examine the frequency distributions of traffic flows from all three reports by dividing the queue counts by the Monitor duration to obtain the values of rate per second.

7.5.2 IMS/ESA MSC Communications Task Trace

MSC message traffic flows can be mapped by the use of the MSC communications task trace. The trace record times can be used to determine whether delays in link performance are attributable to specific functions within the processing cycle. For example:

- **Long AM03 to DM01 times**

  AM03 is the ″what′s next″ trace entry and, if located before a DM01 (setup for write operation), it indicates that a check for output availability has been successful (AM03) and the message segment is to be moved from the message queue buffer to an output buffer (DM01). If the time stamp interval between these two trace entries is abnormally large, there may be a message queue I/O problem, where the message had to be obtained from the message queue data set because of earlier buffer shortages.

- **Long CM02 to DMOI times**

  CM02 is logged when the DDM calls the MSC communications analyzer to get the next output segment of a message through a queue manager internal GN call. DMOI records the entry to the access method. A type X′66′ record is forced to the log by a CHKW call between these two entries. If the time stamp interval between these two records is unusually large, there may be a message queue I/O or WADS problem.

  A comparison of the AM03 to DM01 times with the CM02 to DMOI duration results in either implication or elimination of WADS or message queue I/O as the bottleneck.

7.5.3 MVS-Based Monitors and Traces

There are also several tools for examining MSC performance at the MVS level. These include the MVS Resource Management Facility (RMF), the Generalized Trace Facility (GTF), and TNSTATS from VTAM.

7.5.3.1 Resource Management Facility

Resource Management Facility (RMF) can be used to monitor WADS access rate and response time as well as review the activity to communication devices. Use the RMF I/O Activity Report to obtain an estimate of WADS and message queue data set device service times. Physical transmission time can also be monitored with the I/O Activity Report. If the READ service time is close to the WRITE service time, the logical links on the physical link are close to running out of capacity.

In a sysplex, RMF provides sysplex-wide performance monitoring and capacity planning information, combining performance data for individual systems into sysplex-wide overview reports.

7.5.3.2 Generalized Trace Facility

The Generalized Trace Facility (GTF) shows the data transfer time for channel-attached devices. For timings, start GTF with the TIME=YES parameter. GTF records can be individually listed with the Interactive Problem Control System (IPCS).

7.5.3.3 VTAM TNSTATS

VTAM tuning statistics can be used to determine message rates, the number of path information units (PIUs) sent or received per channel operation, and the total amount of data being transferred.

7.6 MSC Performance Benchmark

The performance measurements were obtained using a 9672 Model R61 processor with a nominal 69 MIPS rate. Although these results may not be achievable in other than this benchmark environment, we trust that showing the relative throughput in various situations will aid in evaluating MSC performance in other environments.

The objectives of the measurements were to:

- Compare the throughput capacity of IMS MSC, CTC, and VTAM-defined links using a single ESCON CTC link, without seriously degrading the terminal response time.
- Compare the transaction rate, processor consumption, and processing path length for work flow traveling through various numbers of MSC links.

The hardware configuration was:

- A 9672 PTS 1 Model R61 with six processors
- Up to 10 ESCON CTCs were available for MSC links
- DFW for databases and the WADS

The systems software environment was:

- MVS/ESA Version 5 PUT Level 9601
- IMS/ESA Version 5 PUT level 9601
- ACF/VTAM V4R3
- Teleprocessing Network Simulator (TPNS) V3R3

The transaction processing environment was as follows:

- TPNS on the P5 processor
- IMSA on the P3 processor to handle input and output without processing any transactions
- IMSB on the P4 processor to handle application scheduling and execution and database access
- Recoverable transactions (therefore, two CHKWs per side)

- Remote system of the MSC partnership has access to two sets of IMS databases on two volumes. The applications perform GU to roots, ISRT, DLET, ISRT, DLET, so that the databases remain in the same state for each benchmark.

Figure 12 illustrates the logical configuration of our benchmark test environment.

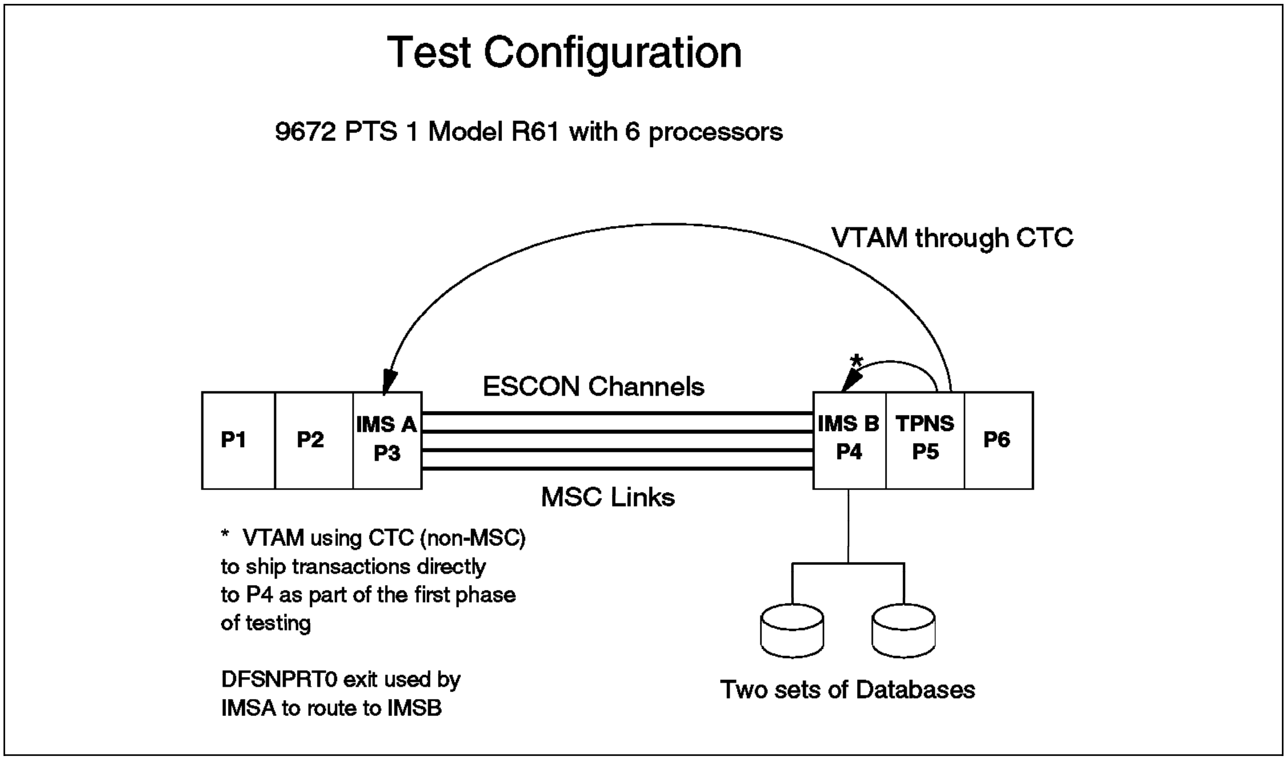


Figure 12. Logical Configuration of Test Environment for Benchmark

TPNS running on processor P5 drives the emulated network workload. The target of the message flow from P5 to P3 is through a VTAM non-MSC ESCON channel connection. The queue counts are allowed to remain in the single-digit range on P3 after the TPNS rate is adjusted on P5.

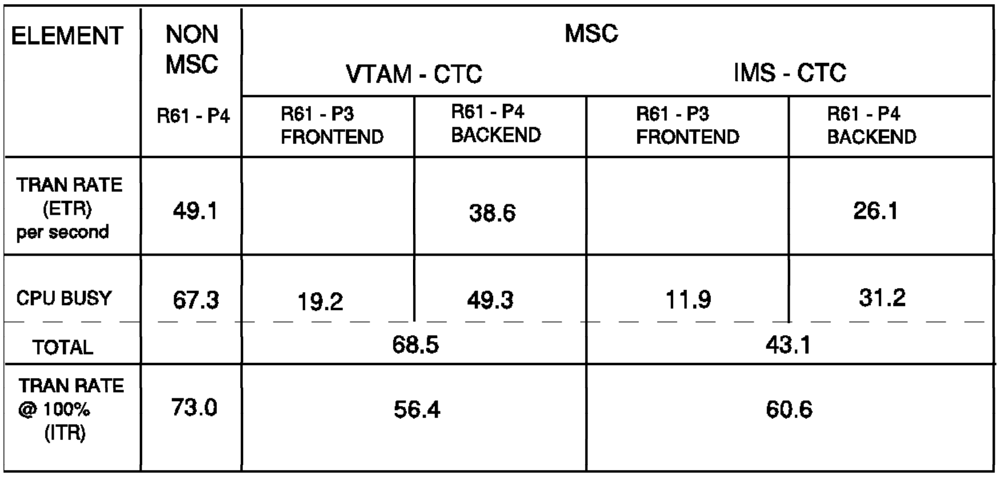
IMS/ESA V5.1 on P3, using user exit DFSNPRT0, routed transactions to P4 first with VTAM MSC ESCON channel link connections, and then with MSC CTC ESCON links. The exit is described in Appendix A.

The Benchmark Steps:

1. The baseline throughput measurement was obtained from direct VTAM flow of messages through an ESCON CTC channel from processor P5 in the PTS complex to processor P4 outside MSC.
2. The environment is allowed to stabilize with P5 driving transactions to IMSA, the queue counts on IMSA staying in the 6 to 8 input range, and transactions being routed successfully to IMSB for execution.
3. A simple checkpoint is taken on both IMSA and IMSB.
4. The benchmark is run for a period of 10 minutes after the environment stabilized.
5. Another simple checkpoint is taken on both systems.
6. The data is reduced.

7.6.1 Results

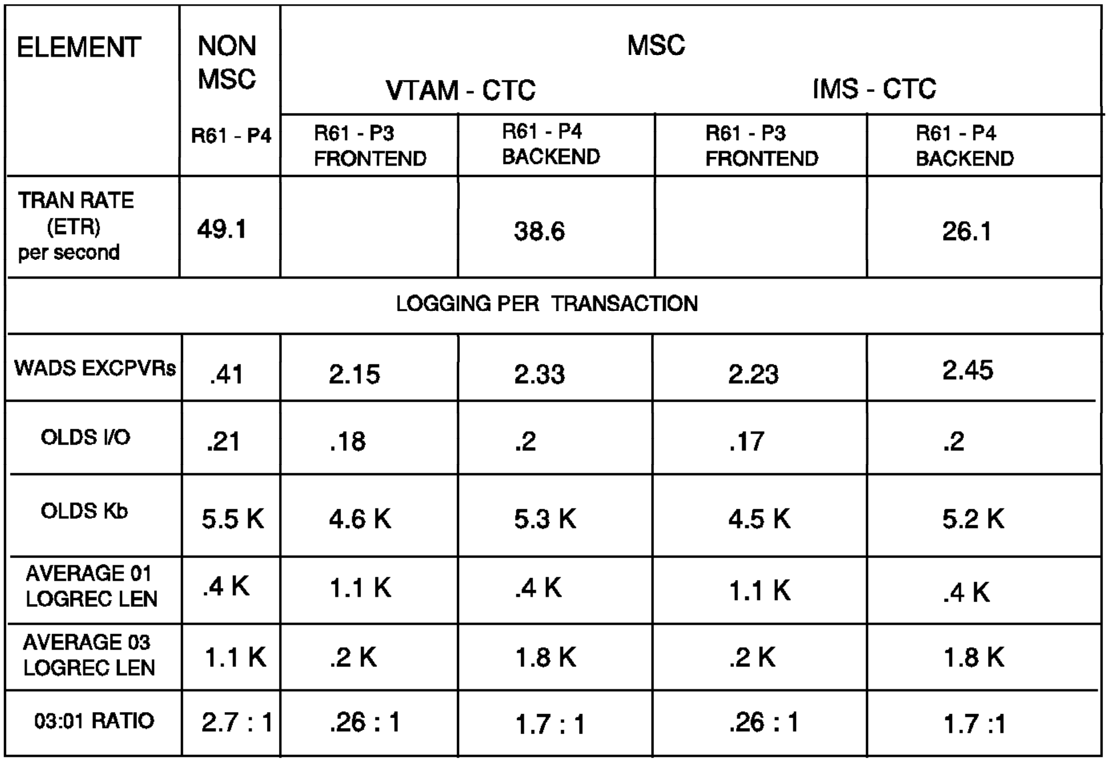
In the first benchmark we established a performance baseline by routing transactions from the TPNS system on P5 directly to P4 with a non-MSC VTAM CTC ESCON connection. Although the rates are higher here, there is none of the fundamental rich support for message routing and user exit interface available with MSC. Figure 13 shows the baseline numbers and the transaction rates and processing costs associated with transaction routing using first a single VTAM MSC CTC link and then a single IMS CTC link.



ELEMENT	NON MSC	MSC			
		VTAM - CTC		IMS - CTC	
	R61 - P4	R61 - P3 FRONTEND	R61 - P4 BACKEND	R61 - P3 FRONTEND	R61 - P4 BACKEND
TRAN RATE (ETR) per second	49.1		38.6		26.1
CPU BUSY	67.3	19.2	49.3	11.9	31.2
TOTAL		68.5		43.1	
TRAN RATE @ 100% (ITR)	73.0	56.4		60.6	

Figure 13. MSC Throughput Measurements with Direct, VTAM, and CTC Connections

The internal throughput rate (ITR) is the transaction rate associated consumption. ITR is calculated by dividing the transaction rate by the total processor % busy value. For example during the IMS-CTC test the total processor busy on the P3 originating and P4 destination processors was 43.1. With a measured rate of 26.1 transactions per second, the ITR value is (26.1/43.1)*100 = 60.6 at 100% processor utilization. The ITR value allows comparisons of throughput from test data obtained from processing at different original processor % busy rates. Figure 14 on page 65 continues to present the results of the first benchmark by repeating the transaction rate per second and then mapping data associated with logging for the test cases.



ELEMENT	NON MSC	MSC			
		VTAM - CTC		IMS - CTC	
	R61 - P4	R61 - P3 FRONTEND	R61 - P4 BACKEND	R61 - P3 FRONTEND	R61 - P4 BACKEND
TRAN RATE (ETR) per second	49.1		38.6		26.1
LOGGING PER TRANSACTION					
WADS EXCPVRs	.41	2.15	2.33	2.23	2.45
OLDS I/O	.21	.18	.2	.17	.2
OLDS Kb	5.5 K	4.6 K	5.3 K	4.5 K	5.2 K
AVERAGE 01 LOGREC LEN	.4 K	1.1 K	.4 K	1.1 K	.4 K
AVERAGE 03 LOGREC LEN	1.1 K	.2 K	1.8 K	.2 K	1.8 K
03:01 RATIO	2.7 : 1	.26 : 1	1.7 : 1	.26 : 1	1.7 :1

Figure 14. Logging Data Associated with the First Benchmark

7.6.1.1 First Benchmark

The VTAM-CTC provides better throughput at 38.6 transactions routed per second compared with 26.1 in the IMS CTC environment on one link.

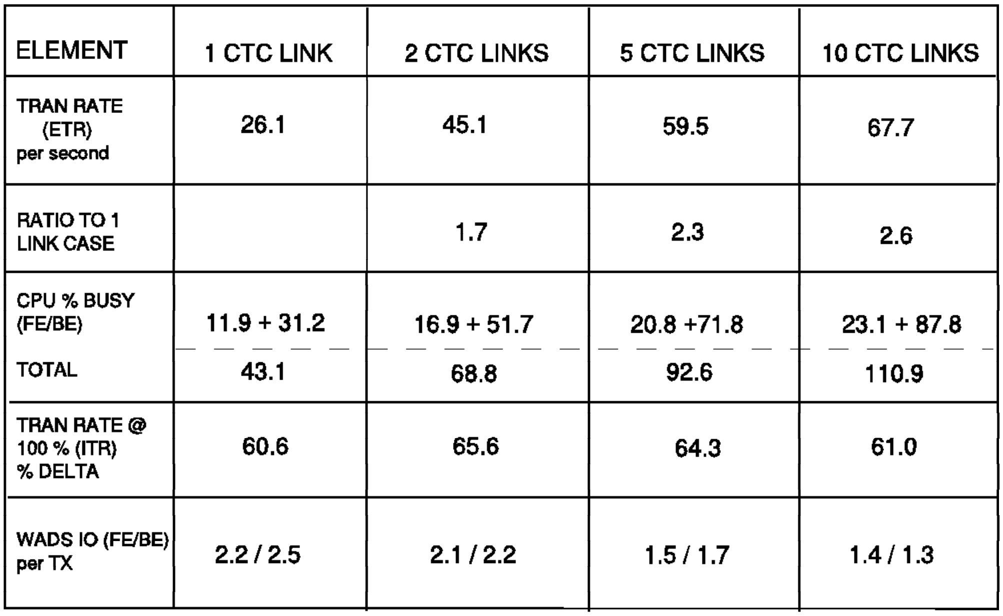
Although the WADS EXCPVRs in the MSC test cases are much higher than without MSC because of the checkwrite calls, neither the VTAM nor CTC MSC link implementation differed very much in its use of the logging resources. If more information about log activity between checkpoints is required, assemble the DSECT for the type X′45′ log record, and look for the X′4507′ logger statistics section. Assemble macro ILOGREC RECID=45 to obtain the log record mapping.

The main MSC contributors to the overhead are:

- A total of four extra WADS I/Os per transaction.
- Delays in link data transfer while the positive response from the receiving side of the link is shipped.
- Double OLDS logging between the IMS partners.
- Increased control task processing as compared to non-MSC because of path length increases.

7.6.1.2 Second Benchmark

Figure 15 presents the results from our second benchmark, which compared throughput rates using various numbers of MSC IMS-CTC links.



ELEMENT	1 CTC LINK	2 CTC LINKS	5 CTC LINKS	10 CTC LINKS
TRAN RATE (ETR) per second	26.1	45.1	59.5	67.7
RATIO TO 1 LINK CASE		1.7	2.3	2.6
CPU % BUSY (FE/BE)	11.9 + 31.2	16.9 + 51.7	20.8 +71.8	23.1 + 87.8
TOTAL	43.1	68.8	92.6	110.9
TRAN RATE @ 100 % (ITR) % DELTA	60.6	65.6	64.3	61.0
WADS IO (FE/BE) per TX	2.2 / 2.5	2.1 / 2.2	1.5 / 1.7	1.4 / 1.3

Figure 15. Throughput Measurements of MSC IMS-CTC Links

Using 10 links resulted in 2.6 times the throughput as compared with using 1 link. Here are the first 2 of 10 sets of MSC Stage 1 statements used to prepare for this benchmark:

Portion of Stage 1 Showing 2 of the 10 Sets of Statements

```
LNK34N0  MSPLINK TYPE=CTC,ADDR=B10,BUFSIZE=4096,DDNAME=CTC34N0
         MSLINK  MSPLINK=LNK34N0,PARTNER=CA,OPTIONS=FORCSESS
MSN34N0  MSNAME  SYSID=(040,030)
*
LNK34N1  MSPLINK TYPE=CTC,ADDR=B11,BUFSIZE=4096,DDNAME=CTC34N1
         MSLINK  MSPLINK=LNK34N1,PARTNER=CB,OPTIONS=FORCSESS
MSN34N1  MSNAME  SYSID=(041,031)
*
* ....
```

The number of WADS I/Os per transaction decreases as the throughput picks up because of efficiencies in logging algorithms handling batched requests for services.

Efficiency improves as the number of links increases, but the leveling off of transaction routing rates as CPU percent busy increases indicates a state of diminishing return on resources above five links.

7.7 MSC Performance Predictions

The benchmarks were completed using connections within the same processor complex. What would be the expected transaction routing rates if we were using remote links or experiencing excessive WADS or message queue I/O? The majority of time spent is going to be for some type of I/O, either to the WADS, queue data sets, or across the link.

Two performance scenarios for MSC VTAM CTC connections for simple front end and back-end environments are mapped below, as well as one extra scenario where there is a delay in the NCP. Table 8 provides the first example of possible accumulated I/O times with no message queue I/Os.

Table 8. I/O VTAM CTC Delays: No MSGQ I/Os to Message Queue Data Sets

Possible I/O Delays	**Average I/O Time (ms)**
Message queue I/Os	0
Checkwrite type 66 log record	4
Send from IMSA to IMSB	2
Checkwrite type 35 enqueue log record	2
Checkwrite type 66 log record	2
Send from IMSB to IMSA	2
Checkwrite type 35 enqueue log record	4
Total time	16

At 16 ms, a maximum of 1000 ms / 16 ms = 62 messages per second could be routed between the two systems. The elapsed time to transmit the positive acknowledgment is not included in this review.

Let us now assume a delay associated with I/O to the MSGQ data sets at a rate of 0.5 MSGQ I/O per message where each I/O takes 20 ms. We now have in Table 9 a much different potential message flow rate.

Table 9. I/O VTAM CTC Delays: MSGQ I/Os to Message Queue Data Sets

Possible I/O Delay	**Average I/O Time (ms)**
Message queue I/O (20 ms X 0.5 = 10 ms)	10
Checkwrite type 66 log record	4
Send from IMSA to IMSB	2
Checkwrite type 35 enqueue log record	2
Checkwrite type 66 log record	2
Send from IMSB to IMSA	2
Checkwrite type 35 enqueue log record	4
Total time	26

The fraction (1000 ms / 26 ms) equals a maximum routing rate of 38 messages per second. The addition of the message queue I/O reduces the potential message rate of flow by 40% as compared to the calculation in Table 8.

Table 10 on page 68 introduces NCP delays of 100 ms associated with MSC VTAM remote connections.

Table 10. I/O VTAM SNA Remote Delays: MSGQ I/Os to Message Queue Data Sets

Possible I/O Delays	**Average I/O Times (ms)**
Message queue I/O (20 ms X 0.5 = 10 ms)	10 milliseconds (ms)
Checkwrite type 66 log record	4
Send from IMSA to IMSB	4
Average NCP delay of transmission to IMSB	50
Checkwrite type 35 enqueue log record	2
Checkwrite type 66 log record	2
Send from IMSB to IMSA	4
Average NCP delay of transmission to IMSA	50
Checkwrite type 35 enqueue log record	4
Total Time	130

The fraction (1000 ms / 130 ms) equals a maximum routing rate of 8 messages per second.

The number of parallel links required to be defined can be derived from these message flow reviews along with estimates of the expected total message flow rate between systems. Monitoring for MSC performance and taking corrective action are thus very important function when managing IMS in a sysplex environment with MSC.

Chapter 8. IMS/ESA Workload Router

In this chapter we describe the IMS/ESA Workload Router (WLR) for MVS/ESA, which operates with the IMS TM software to provide for the transparent routing or balancing of a transaction workload among two or more (up to 32) cloned IMS systems within a parallel sysplex configuration. The IMS/ESA Workload Router makes exploitation of cloning possible by balancing an IMS workload among the cloned IMS systems, although the Workload Router performs with both cloned and noncloned sysplex configurations.

The Workload Router is an independent product that can be ordered separately from IMS/ESA. Our discussion is based on the Version 1 level of this product although we include a section on Version 2.

8.1 Overview

The WLR functions as IMS Transaction Manager user exits to distribute an IMS transaction workload among two or more IMS online systems interconnected through MSC communication links. The WLR architecture does not preclude direct access to any one of the IMS systems by end users. Some of the highlights of this product are:

- Distributes transactions through MSC that originate from network input (SLU2, 3270, and ISC), program-to-program message switches, and APPC/IMS or OTMA clients.
- Offers server options that support both cloned and non-cloned data communication networks.
- IMS user modifications not required to install the product.
- Provides for weighted distribution of transactions. Different server (back-end) systems can receive different portions of the sysplex-wide workload.
- Supports parallel MSC sessions between the router (front-end) and server systems, thereby facilitates MSC link load balancing.
- Automatically recognizes and avoids routing transactions to unavailable server systems or MSC links. It provides for automatic workload reconfiguration in the event of both planned and unplanned outages.
- Supports the concept of transaction affinity. Particular transactions or groups of transactions can be assigned to designated server systems.
- Implements routing surge detection and suppression capabilities when a substantial difference exists between the goal established for a WLR path and the actual percentage of transactions routed to the path.
- Contains an online, real-time administrator interface for monitoring and dynamically updating the WLR configuration.

8.2 Concepts

The WLR makes exploitation of cloned IMS systems possible by balancing an IMS workload among cloned IMS systems. The WLR is composed of two components: the balancer and the administrative interface. The balancer is the WLR component responsible for routing individual IMS transactions throughout the parallel sysplex. The administrative interface is a real-time interface to the balancer that allows a system administrator to both monitor and adjust the WLR configuration.

8.2.1 Configuration

The WLR is customized through the use of macros similar to those used in an IMS system generation. Five macros are used to:

- Begin and terminate the WLR customization macro set
- Define values for the various WLR general options, such as ROUTER= if this IMS system is to be defined as a WLR router or SURGE= to set the surge suppression threshold.
- Define a group of WLR paths. A WLR path represents a destination from a WLR router IMS system to a given WLR server IMS system. In general, a WLR path has a one-to-one relationship with a specific IMS MSC MSNAME defined in the router system IMS system generation. This macro is used to group these individual WLR paths together as a SYSTEM, which is used by the balancer and administrative interface as a unit to generate various displays and manipulate resources associated with the system, such as paths and affinities.
- Define a single WLR path. The macro used for this purpose must be coded immediately after the macro used to define the WLR path group to reflect that this WLR path belongs to the preceding group. The GOAL parameter defines a path goal expressed as a percentage of the router inbound work load to be routed to this path. The aggregate of all designed path goals must equal 100%.
- Define an affinity destination for one or more transactions or transaction classes.

8.2.2 Balancing with Routers and Servers

WLR superimposes the roles of router and server systems on the IMS systems included in the WLR configuration. Figure 16 on page 71 depicts a three-way cloned IMS system configuration with WLR installed.

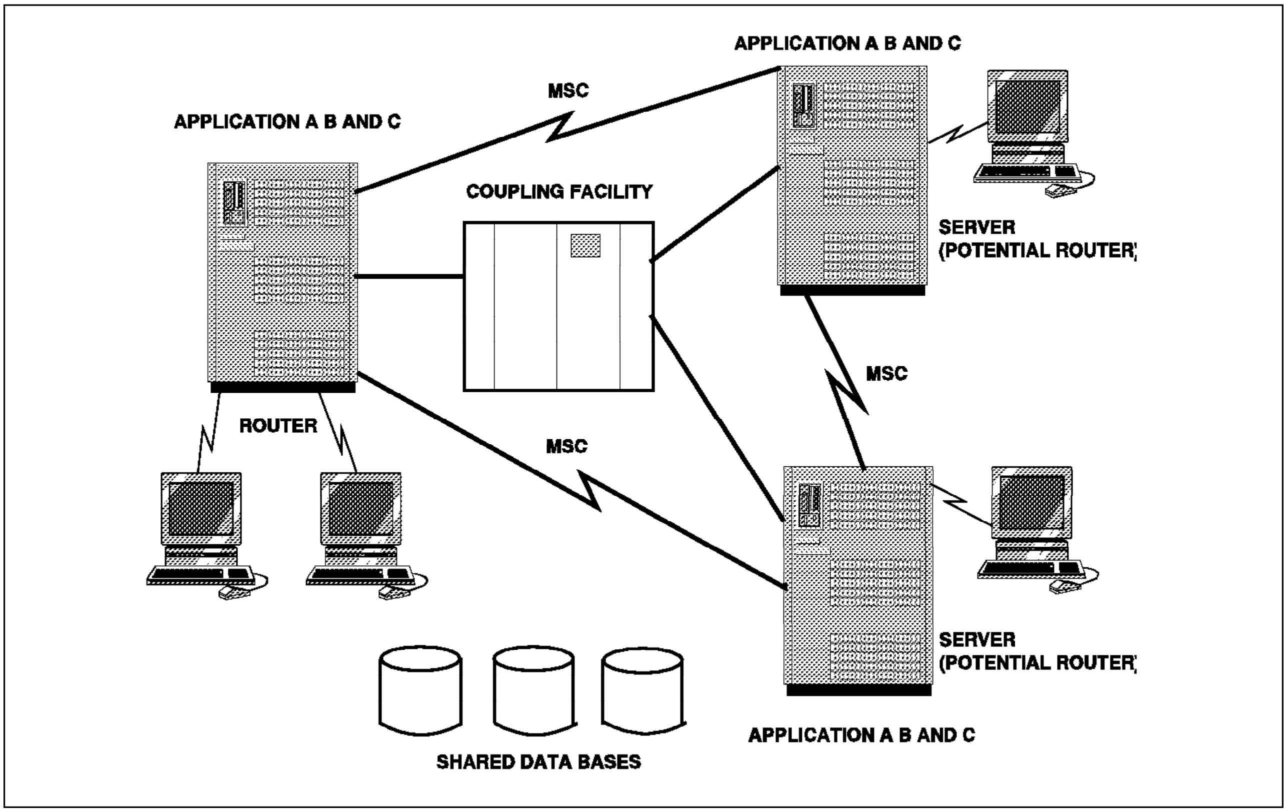


Figure 16. WLR Three-Way Cloned Configuration

A WLR router system is a WLR-capable IMS system that is responsible for routing some portion of the inbound transaction workload to MSC-connected IMS server systems as defined in its WLR customization parameters. The presence of WLR customization parameters and WLR load modules in the IMS system′s RESLIB is what determines whether or not a system is considered to be a router system.

A server-only IMS system has a minimal amount of WLR functionality installed, and its function is to execute WLR-routed transactions.

A given WLR configuration may consist of one or more router systems and one or more server systems. The WLR can also perform in an environment where interconnected WLR router systems play the role of server to one another. This allows for the implementation of transaction swapping. End users connected to either IMS system, entering a variety of different transaction types, could be routed to either system according to affinity to partition application workloads.

8.2.3 Administrative Interface

The WLR administrative interface monitors the status of the WLR router and modifies the WLR configuration dynamically. This interface executes as a standard IMS MPP and uses IMS MFS to interact with the administrator. Any changes made to the WLR configuration through the administrative interface persist until they are again altered by the administrative interface or the IMS control region terminates. To make permanent changes to the WLR configuration it is necessary to recustomize WLR and restart the IMS system. Table 11 on page 72 presents the functions available through the WLR administrative interface.

Table 11. WLR Administrative Interface Functions

Function	Description
ADD	Dynamically adds an affinity entry to the WLR configuration
AFFIN	Displays the entire list of mask or class affinities defined to WLR for a given router system
DISAB	Dynamically disables a variety of WLR resources such as path, affinity masks or class, terminal program or affinity routing
ENABL	Dynamically enabled a variety of WLR resources
GOALS	Alters the path goals established through WLR customization. GOAL is the percentage of input workload to be routed to a path
GRUSM	Obtains an overall graphic view of the WLR-defined systems, their associated goals, and actual routing statistics.
LEVEL	Manually resets the WLR routing statistics to prevent a possible routing surge
MENU	Presents a menu of all WLR administrative functions
PARMS	Displays a list of WLR parameter settings
PFKS	Displays a list of program function key (PFK) definitions for the WLR administrative interface
ROUTE	Dynamically routes affinity entries to alternative destinations
RROUTE	Dynamically reroutes affinity entries to the original destination previously altered with the ROUTE function
QLCL	Modifies the affinity queue setting to QUE=LCL. When certain conditions are met, the WLR router redirects the input message to the local path.
QOFF	Modifies the affinity queue setting to QUE=OFF
QON	Modifies the affinity queue setting to QUE=ON. When certain conditions are met, the WLR router ignores the path status and queues the message to the path.
SET	Modifies the general WLR parameters established through the WLR customization process.
SHOW	Displays the resources and status associated with a particular WLR system or a particular path
STATS	Displays a list of the WLR systems and their related paths

You can see that the administrative interface transactions provide considerable control over the WLR environment.

8.3 WLR IMS User Exit Support

When installed, WLR has direct interaction with the three IMS exit routines listed below. WLR does not directly replace any of these exits in IMS systems, except DFSCMPR0 (on server systems only).

- **DFSPPUE0** (Partner Product Initialization exit)

  WLR uses DFSPPUE0 before IMS is ready to start up (before the DFS994I start-complete message is issued) to perform WLR initialization processing on the router system. Other products that run with IMS can still use DFSPPUE0 because WLR attempts to dynamically load a module with the external module name, LCLPPUE0, during its processing. If the module is present in the IMS RESLIB, WLR passes control to the LCLPPUE0 module, just as IMS passes control to DFSPPUE0.

- **DFSCMTR0 or DFSNPRT0** (IMS limits the use to one or the other)

If either exit is used in a WLR router system, it is given control before any routing decision has been made by WLR. Either exit may influence the WLR router′s decision by changing the transaction destination name because the changed name is used by the router during affinity routing lookup processing.

- **DFSCMPR0**

  In a WLR router system, DFSCMPR0 is given control before any routing decision has been made by WLR. WLR router decisions can be influenced by nonzero return code settings in register 15 and proper MSC SYSID or MSNAME selections on exit from DFSCMPR0.

8.4 IMS/ESA WLR Version 2

Version 2 of the WLR has undergone significant enhancements that are valuable in a parallel sysplex environment and to a much broader base of IMS installations. WLR V2 introduces the following new features and functions:

- **Remote Destination Routing (RDR)**

  WLR can be used to provide enhanced availability and throughput to users of IMS remote transactions and/or MSC directed routing. The IMS architecture restricts remote transactions to be assigned to only one logical or physical MSC link at a time. Any failure of an MSC link will impact availability. With WLR Version 2 RDR, through its ability to assign remote destinations (transactions and MSNAMES) to a group of MSC links, an MSC link can be eliminated as a single point of failure. If one of the WLR-defined MSC links fails, the WLR automatically avoids routing work to it and uses the alternative links to communicate with partner IMS systems. When the failed MSC link becomes available, WLR automatically reinstates routing to it.

- **Affinity Routing Extensions**

  In support of the RDR enhancements, WLR Version 2 provides extensions to the affinity routing mechanism. To simplify the identification of remote destinations to be managed by RDR, WLR′s affinity routing mechanism has been extended to support the identification of remote destinations by IMS MSC remote SYSID. A single WLR affinity definition can be created to associate an MSC MSNAME and all of its associated remote transactions with a single destination: an individual MSC link or a group (system) of MSC links.

- **System Targeted Affinities (STA)**

  WLR Version 2 users can assign affinity definitions to a system (GROUPING) of WLR defined paths or links. STA offers greater flexibility, availability, and load balancing capability to destinations with defined affinity.

- **Improved Configuration Management**

  Configuration management has been improved with Version 2. A new checkpoint function has been added to the administrative interface, enabling WLR system administrators to take a snapshot of the WLR configuration at any given point in time, including real-time changes made to the configuration through the interface. A WLR post-processing utility may later be used to re-create the WLR configuration macros from the checkpoint data.

- **Diagnostic Tracing**

A new diagnostic trace has been added to WLR. Once activated, the WLR records trace data on the IMS system log. The trace records contain data that can be used to assist in diagnosing WLR suspect problems.

- **New Routing Controls**

  Through a new global routine inhibitor control, WLR now can optionally route transactions defined to IMS as FAST PATH potential. This change provides more granular control to the WLR administrator in managing the WLR configuration.

8.5 Additional Information

The product number for the Workload Router, which has been generally available since May 31, 1996, is 5697-074.

The product number for the Workload Router Version 2 is 5697-B87. Version 2 was announced on December 3, 1996, with a planned availability date of February 1997.

The following material is available and provides more detailed information:

- *IMS/ESA Workload Router User Guide* SC26-8945-01
- *IMS/ESA Workload Router Licensed Program Specification* GC26-8946-00

Chapter 9. IMS/ESA Version 6 MSC Enhancements

IMS/ESA Version 6 enhances the facilities available for sharing messages across many different IMS systems in a parallel sysplex. In this chapter we describe the shared queues function of IMS Version 6, concentrating on its use in a sysplex environment with MSC. We cover the following topics:

- Terminology changes
- Message routing in a shared queues environment
- MSC APPC and OTMA support
- /MSVERIFY support
- Improvements in sysgen limits for MSC resources
- UTC time support
- Removal of BSC link support
- Compatibility items

9.1 Terminology Changes

Front-End and Back-End IMSs within a Shared Queues Group

A shared queues group (SQG) is a number of IMS Version 6 systems that share a common message queue structure located in the coupling facility. The IMS systems that make up an SQG are referred to as front-end and/or back-end systems. The front-end system is the message-receiving network or MSC input link system. The back-end system is the transaction processing or MSC output link system. The back-end system retrieves the messages from the IMS message queue structures on the coupling facility and processes the transaction. The output message is then placed back on the message queue on the coupling facility, ready for the front-end IMS system to send back to the originating terminal or system. If a message stays within one IMS system in an SQG, that system is both the front-end and back-end system.

Remote, Local, and Intermediate Systems

Remote, *local*, and *intermediate systems* are MSC terms and have the same meaning in IMS/ESA Version 6 as they had in previous releases. A message is said to be processed in a local system when the message stays within that system. All IMSs within the same SQG are considered to be local to each other, as the message does not get passed outside the group. The remote system is the final destination IMS where the message is being sent. An intermediate system is a system through which the message is sent to reach the final remote system.

9.2 Message Routing in a Shared Queues Environment

Figure 17 on page 76 displays the MSC interface between systems that are in an SQG and other IMS systems.

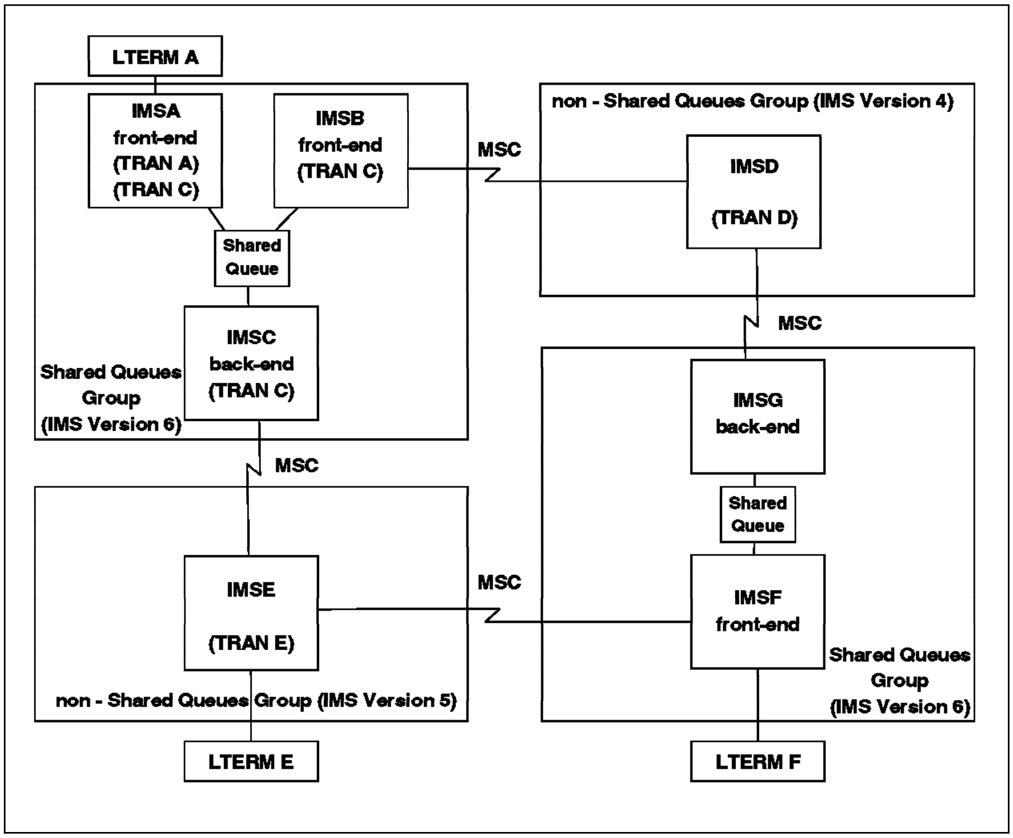


Figure 17. Message Routing in an MSC Shared Message Queue Environment

IMSA, IMSB, and IMSC are defined to an SQG. All messages are inserted to the shared queues, eliminating the need to have MSC messages routing between IMSs in the SQG. Using the configuration in Figure 17, we map the flow of message routing using various connection paths:

- **Local transaction in an MSC front-end system**

  If LTERMA enters a message for TRANA, it will be placed on the shared transaction ready queue for TRANA. Because IMSA has registered an interest in this queue, the transaction can be processed in the IMSA front-end system. A *queue* is a shared queues term for a transaction or lterm.

- **Local transactions in an MSC front-end and back-end system**

  LTERMA enters a message for TRANC. This transaction has been defined in IMSA, IMSB, and IMSC. IMSA places the message on the shared transaction ready queue for TRANC, where it can be processed by IMSA, IMSB, or IMSC.

- **Remote transaction in MSC shared queue front-end system sent out through a back-end system to a remote non-shared-queue system**

  If LTERMA enters a message for TRANE, it is placed on the shared transaction ready queue for TRANE. IMSC sends the message through the MSC link to IMSE for processing. The response message back to LTERMA is sent from IMSE across the MSC link to IMSC. IMSC places the message on the shared lterm ready queue for LTERMA. IMSA takes the message off the shared lterm ready queue and sends it to LTERMA.

- **Remote transactions from non-shared-queue IMS to a shared-queue IMS**

  If LTERME in IMSE enters a message for TRANA, the message is placed on the message queue in IMSE and sent across the MSC link to IMSC. IMSC places the message on the shared transaction ready queue for TRANA, where it is retrieved by IMSA and processed. MSC is not used to move messages between the IMS members of the SQG.

- **Remote transaction from a non-shared-queue IMS, through a shared queue IMS intermediate system, to a remote non-shared-queue IMS**

  LTERME on IMSE enters TRAND. The message is sent across the MSC link to IMSF where it is determined that the destination is remote. The message is inserted to the shared remote queue, pulled off the shared queue by IMSG, and sent to IMSD. On completion of the transaction on IMSD, the response is sent back to IMSG, recognized as remote in IMSG, and inserted to the shared remote queue. The message is pulled off by IMSF and sent across the MSC link to IMSE where it is sent to LTERME.

To support MSC in a shared queues environment, the MSC feature must be sysgened in all IMSs in the SQG. There is no check to enforce this at IMS initialization, or when new IMSs join the SQG. It must be set up beforehand. If the MSC feature is not sysgened in all IMSs, message routing will not work. To sysgen MSC, define at least one MSC link (that is, MSPLINK, MSLINK, and MSNAME macros). Also, if the partner IMS is in the same SQG, link restart will fail for the partner with this message:

```
DFS2149I PARTNER IN SAME SHARED QUEUES GROUP - RESTART ABORTED
```

To support noncloned IMSs within an SQG, IMS creates dynamic LNBs (MSNAMES) and RSMBs (remote transactions) to represent the real LNB and RSMB to the SQG. This is necessary in order to insert and route messages to and from these resources within the SQG.

As IMS systems join and leave the SQG, MSC performs an internal table rebuild to reflect a common view of the routing of all of the IMSs in the SQG. This ensures that there is an accurate mapping of those IMSs to allow them to perform as one single local IMS to the SQG or one single intermediate or remote IMS to IMSs outside the SQG.

9.3 MSC APPC and OTMA Support

APPC and OTMA shared queue names are dynamic in a shared queues environment. Consequently, these queue names cannot be known to IMSs outside the SQG. Therefore, messages for APPC and OTMA originating from IMSs outside the SQG require special routing and handling.

9.3.1 Local APPC and OTMA Transactions

Currently, there is an MVS restriction that transactions originating from APPC devices can only execute on the IMS where the APPC session was allocated. Essentially, this is because MVS cannot move the APPC conversation between the front-end IMS and APPC device to the back-end IMS. OTMA support within IMS is a modelled upon the APPC support, so this restriction is extended to OTMA support for IMS Version 6.

9.3.2 Remote APPC and OTMA Transactions

Remote transactions and message switches originating from APPC sessions and OTMA clients do not have this MVS restriction. The message can be sent to a back-end IMS to be then sent out across an MSC link to a remote IMS to process. The response to the APPC device or OTMA client has to be routed back to the source front-end IMS.

9.4 /MSVERIFY Support

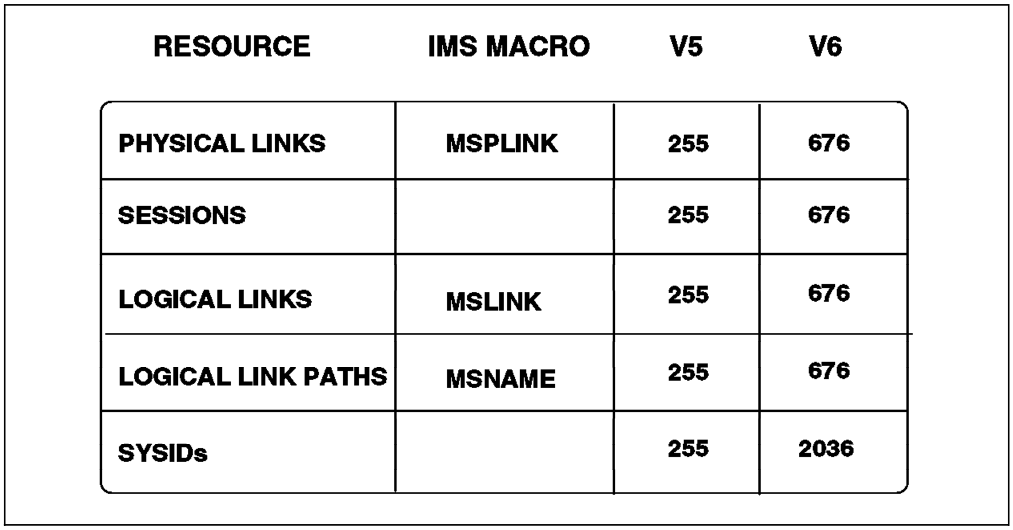
Externally, there is no change to /MSVERIFY support in an IMS/ESA Version 6 shared queues environment. Any IMS in a shared queues environment connected to a remote IMS outside the SQG can issue or receive an /MSVERIFY request to or from the remote IMS. However, IMSs within the SQG that do not have active MSC link connections cannot send or receive /MSVERIFY requests.

9.5 Improvements in the Sysgen Limits for MSC

An IMS subsystem is defined to communicate with other IMS subsystems using MSC through the use of the system generation MSLINK, MSPLINK, and MSNAME macros.

Before IMS/ESA Version 6, the maximum number of IMS systems that could be linked together through MSC was limited to 255. With IMS/ESA Version 6 the number of SYSIDs has increased from 255 to 2036, and the maximum number of physical and logical links has increased from 255 to 676.

With MSC networks growing, and the cloning of subsystems as an inherent part of the sysplex environment, the sysgen limits for MSC associated with IMS/ESA Version 5 are considered to be a major constraint. Now, with the increased number of SYSIDs, logical links, and physical links that Version 6 allows, more IMS systems can be added to the MSC network and more SYSIDs can be used for local and remote transactions and MSNAMEs, thereby increasing MSC network throughput. This enhancement is a positive element in the growth of the use of MSC networks migrating to a sysplex environment. Figure 18 compares the MSC generation limits of IMS/ESA Versions 5 and 6.



RESOURCE	IMS MACRO	V5	V6
PHYSICAL LINKS	MSPLINK	255	676
SESSIONS		255	676
LOGICAL LINKS	MSLINK	255	676
LOGICAL LINK PATHS	MSNAME	255	676
SYSIDs		255	2036

Figure 18. Comparison of IMS/ESA Version 5 and 6 MSC Generation Limits

The figures that follow illustrate single IMS systems connected together through MSC. SQGs of IMS Version 6 systems are not highlighted, but generally an IMS Version 6 SQG could be considered as one IMS Version 6 image to its remote MSC partners. Figure 19 on page 79 demonstrates the use of SYSID values above 255 for multiple IMS/ESA Version 6 systems.

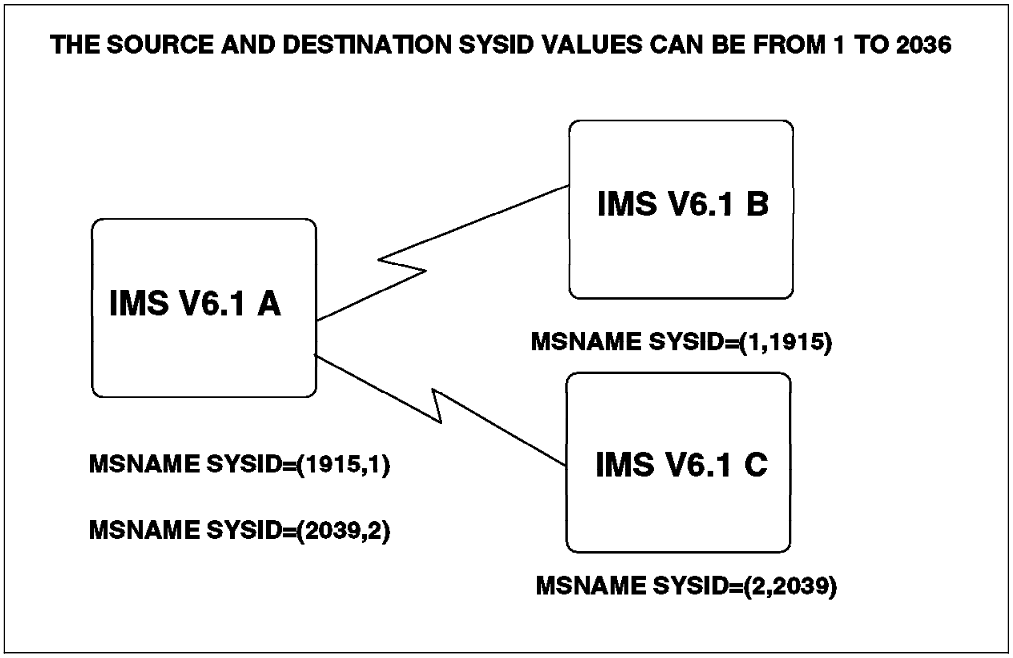


Figure 19. Connection of Multiple IMS/ESA Version 6 Systems with MSC

Communication in an MSC network between an IMS/ESA Version 6 system and a lower-level IMS system can only use SYSIDs in the range of 1 to 255 in *both* systems as specified in the MSNAME and APPLCTN macros.

If the IMS Version 6 system has SYSIDs over 256, it will receive an error message when it attempts to route a transaction to the lower-level IMS. The same situation will occur when an IMS/ESA system lower than Version 6 attempts to route a transaction using MSC to a Version 6 IMS subsystem with SYSIDs between 256 and 2036. Figure 20 shows the connection of an IMS/ESA Version 6 system to an IMS/ESA Version 5 system.

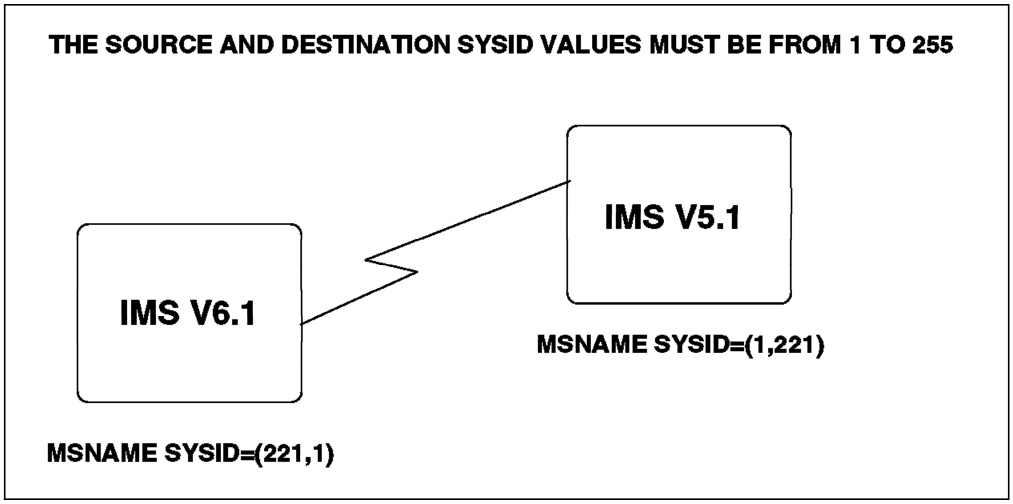


Figure 20. Connection of an IMS/ESA Version 6 System to an IMS/ESA Version 5 System

Figure 21 on page 80 displays the restriction associated with using an IMS/ESA Version 5 system as the intermediate system between two IMS/ESA Version 6 systems.

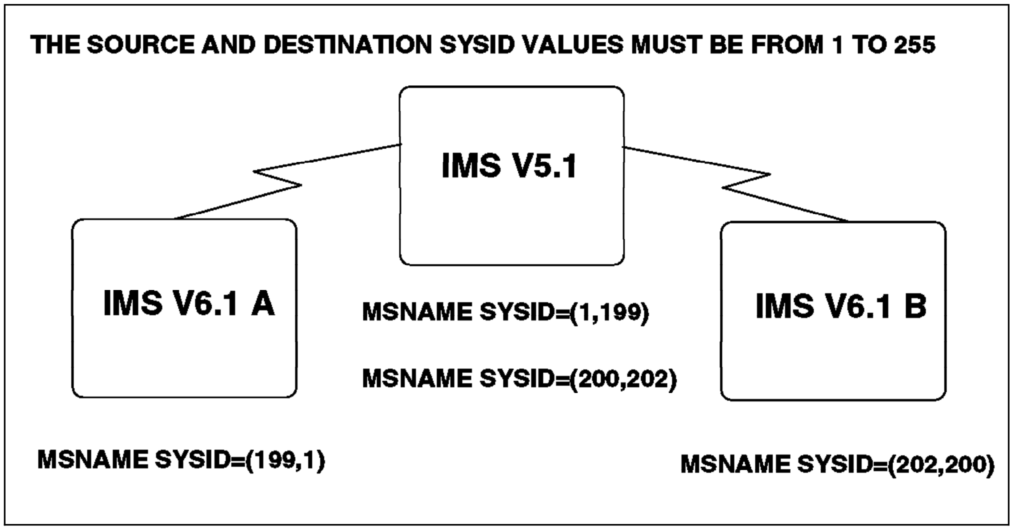


Figure 21. IMS/ESA Version 6 Systems with an Intermediate IMS/ESA Version 5 System

Figure 22 shows the connections between an IMS/ESA Version 6 and Version 5 and Version 6 systems.

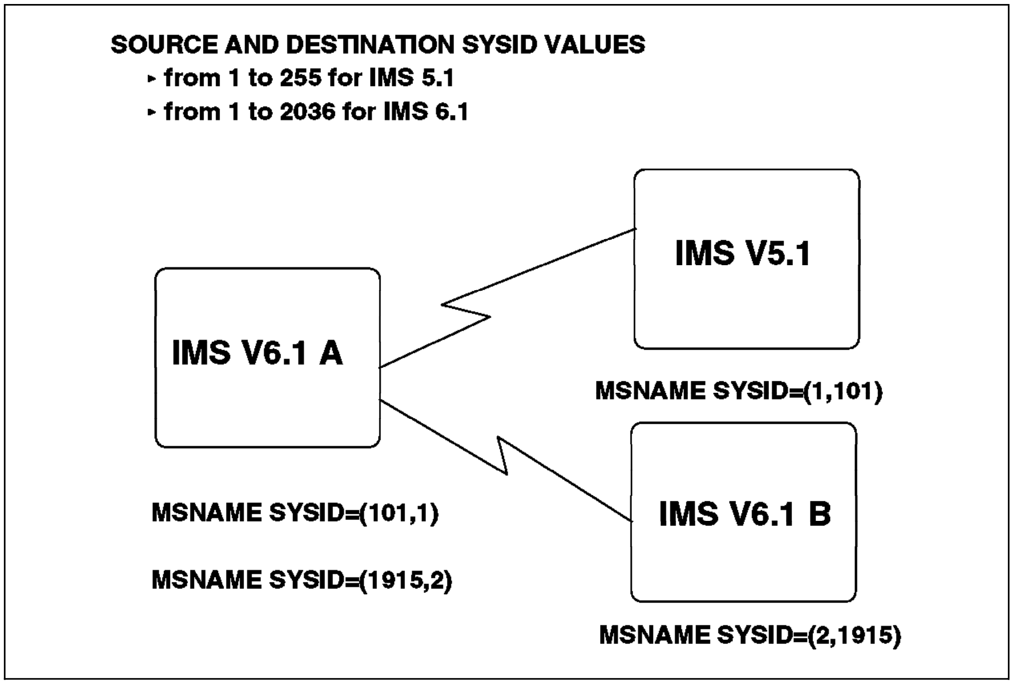


Figure 22. MSC Interface between IMS/ESA Version 6 and IMS/ESA Version 5 and 6

9.6 UTC Time Support

To support the time stamp format change from local time to universal time, coordinated (UTC), MSC calls to timer services have been changed to request the new UTC times. Because IMS/ESA Versions 4 and 5 do not support UTC times, MSC compatibility logic converts from local to UTC times and vice versa in the following situations:

- The partner system is an IMS Version 4 or Version 5 system.
- A message originating in an IMS/ESA Version 4 or 5 system is sent through an intermediate IMS/ESA Version 6 system.

This support has been included internally in IMS/ESA Version 6, so changes are not required from users or applications.

9.7 Removal of BSC Link Support

The sysgen support for link type BSC has been removed from the MSPLINK macro, so BSC link types cannot be generated in IMS/ESA Version 6. Also the MSC BTAM device-dependent modules are not included in the IMS nucleus.

9.8 Compatibility Items

When using shared message queues in IMS/ESA Version 6, there are four compatibility items to consider:

- **TMR prefix segment**

  The MSC fixed prefix has been converted to the transaction router (TMR) prefix. The MSC prefix no longer exists in Version 6, but the MSC extension prefix still does. The TMR prefix is converted to the MSC prefix if the partner is Version 4 or Version 5. This is handled internally, so there is no impact for existing systems.

- **Two-byte SYSIDs to support 2036 SYSIDs**

  In Version 6, SYSIDs have a length of 2 bytes to support the large values. One-byte SYSIDs will be transferred to support IMS/ESA Version 4 and Version 5 but message

  ```
  DFS2134 INVALID SYSID DETECTED LINK x
  ```

  will be issued if the SYSID is greater than 255 and the partner system is either Version 4 or Version 5.

- **UTC times**

  Both the 12-byte and 8-byte times are sent out, but the 12-byte time is converted to an 8-byte length before entering IMS Version 4 and Version 5 partner systems. Also if a message with an 8-byte local time is received by an intermediate Version 6 system, IMS converts the 8-byte time to a 12-byte time.

- **Link restart message size increased**

  The additional fields added to the restart block for link restart in Version 6 have caused incompatibility between IMS/ESA Version 6 and Version 4 because IMS/ESA Version 4 still checks the length of the restart block it receives from another IMS. This check was removed in Version 5, so there are no compatibility issues there. A compatibility APAR (PN91176) will have to be applied to IMS/ESA Version 4 or the following error message will occur at link restart time:

```
DFS2146 INVALID DATA BLOCK RECEIVED LINK xx.
```

Chapter 10. Miscellaneous Considerations

This chapter describes some further non-IMS considerations that you should take into account when planning for the use of MSC. In particular, we look at the use of DB2, processing limits that are placed on transactions, and the use of SYSOUT data sets.

10.1 DB2 Considerations

Most IMS installations run with some DB2 workload in addition to their IMS database workloads. DB2 version 4 supports data sharing. If it is necessary to split your workload over a sysplex before you can implement DB2 V4, you have some options:

- Isolate the DB2 transactions to a specific IMS subsystem by definition. You have to identify all transactions that access DB2 data directly, or have an affinity to a DB2 transaction, and define them to a single IMS. DB2 users must then log on to the IMS that specifically supports the DB2 applications.
- Route DB2 transactions to a specific IMS using MSC or ISC mechanisms. You would still have to identify DB2 transactions in the routing exit, but it would be transparent to users of the DB2 applications.
- You can implement the distributed data facility (DDF) of DB2. DDF enables you to access DB2 data on a remote DB2 subsystem without any application program modifications.

10.1.1 Identifying IMS Transactions That Access DB2 Data

You can identify transactions with DB2 affinities by comparing a list of PSBs defined to IMS with a list of DB2 plan names. Because IMS is the coordinator of the syncpoint processing of application programs under its control, it has to communicate with DB2 at various stages of a program′s execution. Communication between IMS and DB2 is at the DB2 plan level. IMS by default associates a DB2 plan with a PSB name, that is, IMS assumes that the DB2 plan name is the same as the PSB name. Although this is true for most installations, the option exists to override this default by specifying a resource translate table (RTT) that contains a list of PSB names and their associated DB2 plan names. The RTT is an assembled module that is identified through the external subsystem parameter (SSM). IMS allows the specification of an SSM parameter in both the control and dependent regions′ startup JCL that refers to a named member in IMS.PROCLIB. This member contains the name of the DB2 subsystem that can be accessed from applications running in this region, and it can also specify an RTT name.

You can get a list of plans by querying the DB2 catalog. Use the sample SQL query below to get a list of all plans by plan name, the name of the creator, and the bind date, which can be useful for deciding whether the plan is still valid.

DB2 Catalog Query

```
Select Name, Creator, Binddate, Valid from SYSIBM.SYSPLAN
```

Compare the output of this query with a list of PSBs to see which transactions can access DB2. The accuracy of this comparison depends very much on your installation, however. If you create a DB2 plan by default when a new PSB is created, this check is useless. It is also inaccurate if you have multiple transactions per PSB and use the program with the same name as the PSB just as a router to different programs. The only accurate way of determining DB2 presence in IMS programs executed using a specific PSB is to scan the IMS log.

For pure batch programs, the DDITV02 data sets must be checked.

For more information, see the *DB2 for MVS/ESA V4 Administration Guide*, SC26-3265-00.

10.2 The PROCLIM Value in Heterogeneous Hardware Environments

The second parameter in the PROCLIM statement in the TRANSACT Stage 1 macro is the amount of time in seconds (ranging from 1 to 65535) allowable to process a single transaction. This number specifies the maximum CPU time allowed for each message to be processed in the message processing region.

The amount of work processed during a CPU second on an ES/9000 processor is not the same as the amount of work processed in a CPU second on a 9672. Therefore the PROCLIM values should be set according to the expected execution level hardware resources. Otherwise applications that run successfully on one processor may abend with code ABU0240 (application program exceeded the allowable execution time) on other processors. To prevent run-away task situations, you might not want to set similar higher values for all IMS systems running on different processor types within the sysplex.

10.3 SYSOUT Data Sets in Parallel Sysplex Environments and MSC

SYSOUT data sets are sequential files that IMS application programs view as terminals. Programs can send messages to these *terminals*, and these messages are written to the SYSOUT data sets in the order of their creation.

When IMS subsystems are cloned, each clone could have its own version of the SYSOUT data sets. In this case, the same terminal definitions would be defined on each IMS system. Of course, each system would have different data sets.

SYSOUT data sets are usually extracted and read by batch programs. It is likely that the multiple SYSOUT data sets will have to be either concatenated or merged. It may not be possible to merge the records in their order of creation. (That would require a sort field in the records, which might not exist.) Putting a time stamp field into the messages could resolve this problem.

MSC can be used as an alternative to the creation and merging of multiple SYSOUT data sets. Because these data sets are treated like terminals, they can be defined as remote MSC terminals.

In an environment where an IMS Subsystem was cloned into IMSA, IMSB and IMSC, IMSA could be defined with the explicit SYSOUT data set and IMSB and IMSC could route messages to the remote destination through MSC links. Then the records would be written to IMSA′s SYSOUT data set in the order of arrival.

There are a few drawbacks to this solution. The order of arrival to IMSA from multiple cloned IMS systems might not be in the order of creation on the originating systems because of delays in sending the messages between systems. Also, if IMSA fails, the messages for the SYSOUT data set will remain in the message queues of IMSB and IMSC until IMSA is restarted and the MSC connections are reestablished.

Appendix A. Sample Link Availability Check Subroutine

The following assembler code could be used in either DFSNPRT0 or DFSCMTR0 to inspect the availability of all resources belonging to a logical link path. The control block fields have been mapped against current IMS/ESA Version 5 control block level macros to ensure accuracy of reference. You should re-check this code carefully if using other releases of IMS.

Here is an overview of the logic of this subroutine:

- Register 2 at entry to the exit is expected to point to the destination name.
- Register 11 on entry to the exit is expected to point to the Systems Contents Directory (SCD).
- Fields SCDSMBEP and SCDNSMB contain the address of the SMB pool and the number of SMBs in the system.
- The destination name in the SMB (SMBNAME) is compared to the field pointed to by Register 2 on entry to the exit. If equal, we fall through to the process described by the following bulleted item. If not, we keep checking all SMBs in the system, and, if we find that our destination is not an SMB, we return from the exit.
- If the SMBNAME in the SMB matches the destination name passed on entry to this subroutine via register 2, then the field SMBSIDR contains the destination SYSID.

  This information enables us to determine whether the transaction is to be assigned to a remote system. Because field SMBSIDR has the remote SYSID, if local and remote SYSIDs are identical, the transaction is not assigned to a remote system, and the remainder of the check routine is skipped. This test occurs when the SMBSIDR is used as an index into the SYSID table, addressed by SCDSID. Each SYSID table entry consists of a pointer to the LNB if this is a remote system or zero if it is local.
- If the SYSID table does have a positive pointer to an LNB, that LNB as mapped by the CNT will point to the LXB at field CNTCTBPT.
- Field LXBVCLBP (and field CTBCLB in the CTB) points to the LLB.
- Once the LLB has been located, the status of the link can be determined. Flags CLB2NOIN and CLB2NOOU in file CLBFLAG2 describe the status of the link. If either flag is on, the link is considered down. Logic that could be inserted after label LINKDOWN could then reroute traffic. The Work Load Router Version 2 Remote Destination Routing (RDR) features provides function to respond to this situation as well.

```
**********************************************************************
*                                                                    *
*          Sample Link Availability Check subroutine                 *
*                                                                    *
**********************************************************************

         USING SCD,R11
         L     R3,SCDSMBEP        Address of the SMB pool
         L     R4,SCDNSMB         Number of the defined SMBs
         USING R3,SMB
LKAV000  EQU   *
         CLC   0(8,R2),SMBNAME    Names match
```

```
           BE    LKAV005            Yes, take this SMB
           LA    R3,SCDSMBLN(R3)    Take the next SMB
           BCT   R4,LKAV000         Keep trying till SMBs exhausted
           B     NOTSMB             Destination is not a SMB
LKAV005    EQU   *
           SR    R4,R4              Clear Working Register 4
           IC    R4,SMBSIDR         Pick the Remote SYSID
           DROP  R3
           L     R6,SCDSID          Get the SYSID Table
           MH    R4,SCDSIDL         Multi SID # by length of each
*                                   SID entry which is 4 bytes
           L     R5,0(R4,R6)        Get pointer to LNB
           LTR   R5,R5              Is there a pointer?
           BZ    NOTRMT             No, it is local so leave
           USING CNT,R5             LNB is mapped by CNTs
           L     R4,CNTCTBPT        Get the LXB from the CNTCTBPT
           DROP  R5
           USING CTB,R4             LXB is mapped by the CTB
           L     R5,CTBCLB          Get the LLB into R5
           DROP  R4
           USING IECTDECB,R5        LLB is mapped by the CLB
           TM    CLBFLAG2,CLB2NOIN  Input stopped flag
           BO    LINKDOWN           The link is down
           TM    CLBFLAG2,CLB2NOOU  Output is stopped
           BO    LINKDOWN           The link is down
LKAV010    EQU   *
* Any code to be processed when the link is available
           B     RET
NOTSMB     EQ    *
* The message is not destined for a transaction
           B     RET
NOTRMT     EQU   *
* The transaction is not assigned as a remote
           B     RET
LINKDOWN   EQU   *
* Logic to reroute traffic in case of inoperable link situation
           B     RET
* constants etc.
           LTORG
           ISCD  SCDBASE=0
           IAPS  SMBBASE=0
           ICLI  CLBBASE=0
           ICLI  CNTBASE=0
           ICLI  CTBBASE=0
```

Appendix B. Sample Input Message Routing Exit

Listed below is the JCL stream to assemble and link-edit DFSNPRT0. Included is the source program used for the MSC 10 CTC link performance run on the second benchmark, as discussed in Chapter 7.

After determining whether or not the message originated from an LU6.2 device (which it did not in our performance test), the last digit in the input logical unit (LU) name is checked and the corresponding MSC MSNAME and SYSID are set before we return. For example, input from the LU name ending in 2 would have the MSC MSNAME set to MSN34N2 and the MSC REMOTE SYSID=42. In this way, the message routing is distributed through the index value in the input LU, and, because TPNS in our performance test creates the output message flow in a round-robin manner, the link traffic distribution will be equally distributed.

```
//LNKNPR10 JOB (I51MSC,′A=M12D′),
// I51MSC,
// TIME=99,CLASS=A,
// MSGCLASS=A
//ASM      EXEC PGM=IEV90,REGION=4096K,
//             PARM=′OBJECT,DECK,NOLOAD,NODBCS,LIST′
//SYSLIB   DD DSN=IMSPERF.IMS510.OPTIONS,DISP=SHR
//         DD DSN=IMSPERF.IMS510.MACLIB,DISP=SHR
//         DD DSN=SYS1.MACLIB,DISP=SHR
//         DD DSN=SYS1.MODGEN,DISP=SHR
//SYSPUNCH   DD DISP=OLD,
//         DSN=IMSPERF.IMS510.MOBJDSET(DFSNPRT0)
//SYSPRINT DD SYSOUT=A,
//         DCB=(BLKSIZE=605),
//         SPACE=(605,(100,50),RLSE,,ROUND)
//SYSUT1   DD UNIT=SYSDA,DISP=(,DELETE),
//         SPACE=(CYL,(244,15))
//SYSIN    DD *
*********************************************************************** 00010000
*                                                                     * 00020000
*  MODULE NAME: DFSNPRT0                                              * 00030000
*                                                                     * 00040000
*  DESCRIPTIVE NAME:  INPUT MESSAGE ROUTING EXIT SAMPLE               * 00050000
*                                                                     * 00060000
*********************************************************************** 00070000
*                                                                     * 00070100
*         Licensed Materials - Property of IBM                        * 00070200
*                                                                     * 00070300
*         ″Restricted Materials of IBM″                               * 00070400
*                                                                     * 00070500
*         5695-176 (C) Copyright IBM Corp. 1991                       * 00070600
*                                                                     * 00070700
*********************************************************************** 00160000
*                                                                     * 00170000
*  STATUS:                                                            * 00180000
*                                                                     * 00190000
*  NOTES:                                                             * 00200000
*                                                                     * 00210000
*    DEPENDENCIES:  NONE                                              * 00220000
*                                                                     * 00230000
*    RESTRICTIONS:  NONE                                              * 00240000
```

```
*                                                                     * 00250000
*                                                                     * 00260000
*                                                                     * 00270000
*  MODULE TYPE:                                                       * 00280000
*                                                                     * 00290000
*    PROCESSOR:  ASSEMBLER                                            * 00300000
*                                                                     * 00310000
*    ATTRIBUTES:  REENTRANT                                           * 00320000
*                                                                     * 00330000
*  ENTRY POINT: DFSNPRTO                                              * 00340000
*                                                                     * 00350000
*    PURPOSE:  SEE FUNCTION                                           * 00360000
*                                                                     * 00370000
*    AMODE: 31                                                        * 00380000
*    RMODE: ANY                                               @PN67700 00390000
*                                                                     * 00400000
*  FUNCTION:   THIS EXIT HAS THE SAME FUNCTION AS THE MSC TERMINAL    * 00410000
*              ROUTING EXIT ROUTINE (DFSCMTRO). EITHER EXIT WILL BE   * 00420000
*              CALLED BUT NOT BOTH.  IF BOTH DFSNPRTO AND DFSCMTRO    * 00420300
*              EXIST, ONLY DFSNPRTO WILL BE CALLED (SEE TABLE BELOW). * 00420330
*                                                                     * 00420370
*              IT CAN CHANGE THE DESTINATION NAME OF AN INPUT MESSAGE * 00420400
*              TO ANY LOCAL TRANSACTION OR LTERM DESTINATION.  IN A   * 00420440
*              MULTIPLE SYSTEM COUPLING (MSC) ENVIRONMENT, IT MAY     * 00420480
*              CHANGE THE DESTINATION NAME TO ANY LOCAL OR REMOTE     * 00420500
*              TRANSACTION OR LTERM.  FOR TRANSACTION TYPE MESSAGES   * 00420550
*              IT MAY ALSO OVERRIDE THE MSC DESTINATION SYSTEM        * 00420600
*              IDENTIFICATION (SID) THAT IS CURRENTLY ASSIGNED TO THE * 00420800
*              TRANSACTION IN THE INPUT SYSTEM, AND CAUSE THE MESSAGE * 00421000
*              TO BE REROUTED TO A DIFFERENT IMS MSC SYSTEM.          * 00421400
*                                                                     * 00421700
*              THIS EXIT IS CALLED UPON RECEIVING MESSAGES FROM THE   * 00421740
*              FOLLOWING PROGRAMS/DEVICES/CLIENTS:                    * 00421780
*                                                                     * 00421800
*                - ANY IMS SUPPORTED TERMINAL                         * 00421860
*                - ANY INTERSYSTEM COUPLING (ISC) LU6.1 DEVICE OR     * 00421900
*                  PROGRAM                                            * 00421940
*                - ANY ADVANCED PROGRAM TO PROGRAM (APPC) LU6.2       * 00421980
*                  DEVICE OR PROGRAM, EXCEPT FOR MESSAGES DESTINED      00422000
*                  DIRECTLY TO AN APPLICATION PROGRAM VIA THE           00422060
*                  COMMON PROGRAM INTERFACE COMMUNICATION               00422100
*                  (IE: MESSAGES DESTINED TO CPI-C DRIVEN APPLICATION   00422150
*                  PROGRAMS).                                           00422190
*                - ANY OPEN TRANSACTION MANAGER ACCESS (OTMA) CLIENT  * 00422200
*                                                                     * 00422270
*              FOR LU6.2/ADVANCED-PROGRAM-TO-PROGRAM (APPC) INPUT     * 00422300
*              MESSAGES DESTINED TO A LOCAL OR REMOTE TRANSACTION,    * 00422350
*              OR LTERM, THIS EXIT IS CALLED WHEN THE APPC            * 00422400
*              CONVERSATION IS ALLOCATED AND BEFORE THE DATA IS       * 00422700
*              RECEIVED.                                              * 00423000
*                                                                     * 00423400
*              FOR OPEN TRANSACTION MANAGER ACCESS (OTMA) INPUT       * 00423700
*              MESSAGES, THIS EXIT IS CALLED WHEN THE TRANSACTION     * 00424000
*              MESSAGE IS RECEIVED FROM THE OTMA CLIENT.              * 00424400
*                                                                     * 00424800
*              FOR ALL OTHER INPUT MESSAGES FROM IMS SUPPORTED        * 00425000
*              PROGRAMS AND DEVICES, THIS EXIT IS CALLED WHEN THE     * 00425500
*              FIRST SEGMENT OF THE MESSAGE IS RECEIVED FROM THE      * 00425800
*              DEVICE OR PROGRAM.                                     * 00426000
```

```
*                                                                     * 00426500
*              IN ALL CASES, THE DESTINATION OF THE MESSAGE HAS NOT   * 00426800
*              YET BEEN DETERMINED AND MAY BE CHANGED BY THIS EXIT.   * 00427000
*              FOR ALL MESSAGES DESTINED TO A TRANSACTION, THE        * 00428000
*              DESTINATION MSC SYSTEM MAY BE SPECIFIED.  THE          * 00430000
*              TRANSACTION STILL MUST BE DEFINED (EITHER LOCAL OR     * 00431000
*              REMOTE) IN THE INPUT SYSTEM.                           * 00433000
*                                                                     * 00435000
*              THIS EXIT IS NOT CALLED FOR INPUT FROM A CPI-C DRIVEN  * 00437000
*              PROGRAM OR INPUT FROM A MULTIPLE SYSTEM COUPLING (MSC) * 00438000
*              LINK.                                                  * 00440000
*                                                                     * 00440500
*              FOR TRANSACTION DESTINED MESSAGES, THE EXIT MAY RETURN * 00441000
*              A MSC MSNAME OR SYSTEM IDENTIFICATION (SID). IF THE    * 00441700
*              MULTIPLE SYSTEMS COUPLING (MSC) FEATURE IS SYSGENED,   * 00442000
*              THIS WILL CAUSE THE MESSAGE TO BE REROUTED TO THE      * 00442900
*              REMOTE IMS SYSTEM VIA THE MSC LINK ASSIGNED TO THE     * 00443000
*              MSNAME OR SID. THIS PERMITS TRANSACTIONS WHICH ARE     * 00444000
*              DEFINED AS LOCAL (OR REMOTE) TO BE REROUTED TO A       * 00444700
*              DIFFERENT IMS SYSTEM FOR PROCESSING, THAN WHERE THE    * 00445000
*              TRANSACTION IS CURRENTLY ASSIGNED.  THE TRANSACTION    * 00445800
*              MUST ALSO BE DEFINED AS LOCAL WITH THE SAME            * 00446000
*              CHARACTERISTICS IN THE REROUTED-TO IMS SYSTEM. THIS    * 00447000
*              REROUTE CAPABILITY CAN BE USED FOR TRANSACTION LOAD    * 00447600
*              BALANCING ACROSS IMS MSC SYSTEMS THAT ARE CONNECTED    * 00448000
*              BY MSC LINKS.                                          * 00448800
*                                                                     * 00449000
*              FOR LU6.2 INPUT MESSAGES, THIS EXIT MAY CHANGE THE     * 00450000
*              DESTINATION TO A LOCAL OR REMOTE TRANSACTION BUT NOT   * 00460000
*              TO A LOCAL OR REMOTE LTERM OR TO A CPI-C DRIVEN        * 00465000
*              APPLICATION PROGRAM.                                   * 00470000
*                                                                     * 00475000
*              THE ROUTING TECHNIQUES AVAILABLE ARE:                  * 00480000
*                                                                     * 00490000
*               * CHANGE THE DESTINATION NAME IN THE AREA POINTED TO  * 00500000
*                 BY PARMLIST+16.  THIS CAUSES THE MESSAGE TO BE      * 00510000
*                 ROUTED TO THE NEW DESTINATION WITH NO CHANGE IN THE * 00520000
*                 ORIGINAL TRANSACTION CODE OR DATA.                  * 00530000
*                                                                     * 00540000
*               * CHANGE THE DESTINATION NAME IN THE SEGMENT POINTED  * 00550000
*                 TO BY PARMLIST+12 (IF SEGMENT IS PASSED TO EXIT).   * 00550400
*                 THIS DOES NOT CAUSE REROUTING BUT CHANGES THE       * 00550800
*                 TRANSACTION CODE PASSED TO THE USER APPLICATION     * 00551000
*                 PROGRAM.                                            * 00551700
*                                                                     * 00552000
*               * CHANGE THE DESTINATION NAME IN BOTH THE AREA        * 00552600
*                 POINTED TO BY PARMLIST+16 AND IN THE INPUT SEGMENT  * 00553000
*                 POINTED TO BY PARMLIST+12. THIS ROUTES THE MESSAGE  * 00553400
*                 TO THE NEW DESTINATION AND CHANGES THE TRANSACTION  * 00553900
*                 CODE PASSED TO THE USER APPLICATION PROGRAM.        * 00554000
*                                                                     * 00554700
*               * RETURN A MSNAME OR SID IN THE AREA POINTED TO BY    * 00555000
*                 PARMLIST+32 OR +36.  THIS CAUSES THE MESSAGE TO     * 00555600
*                 BE REROUTED TO THE IMS MSC SYSTEM WHICH THE         * 00556000
*                 ASSIGNED MSNAME OR SID POINTS TO.                   * 00556500
*                                                                     * 00557000
*               * RETURN A MSNAME OR SID, AND CHANGE THE DESTINATION  * 00557800
*                 NAME POINTED TO BY THE PARMLIST AND/OR IN THE       * 00558000
*                 MESSAGE SEGMENT.  THIS WILL CHANGE THE              * 00558600
```

```
*                 DESTINATION NAME THE MESSAGE IS ROUTED TO AND ROUTE * 00559000
*                 THE MESSAGE TO THE SELECTED IMS MSC SYSTEM.         * 00559500
*                                                                     * 00560000
*          E ]                                                        * 00570000
*          X ]                                                        * 00580000
*          I ]                                                        * 00590000
*          T ]     DFSNPRTO     DFSCMTRO     BOTH EXITS               * 00600000
* MSG TYPE   ]      EXISTS       EXISTS         EXIST                 * 00610000
*--------------------------------------------------------             * 00620000
*            ]                                                        * 00630000
*            ]                                                        * 00640000
*   LU 6.2   ]      CALL                        CALL                  * 00650000
*   MESSAGE  ]     DFSNPRTO      NONE         DFSNPRTO                * 00660000
*            ]                                                        * 00670000
*            ]                                                        * 00680000
*            ]                                                        * 00690000
*   OTMA     ]      CALL                        CALL                  * 00691500
*   MESSAGE  ]     DFSNPRTO      NONE         DFSNPRTO                * 00693000
*            ]                                                        * 00694500
*            ]                                                        * 00696000
*            ]                                                        * 00697500
* OTHER TYPE ]      CALL         CALL            CALL                 * 00700000
*   MESSAGE  ]     DFSNPRTO     DFSCMTRO       DFSNPRTO               * 00710000
*            ]                                                        * 00720000
*                                                                     * 00840000
*   ENTRY INTERFACES :                                                * 00850000
*                                                                     * 00860000
*              REGISTERS AT ENTRY :                                   * 00870000
*                                                                     * 00880000
*              R1     = ADDRESS OF STANDARD IMS USER EXIT PARMLIST    * 00890000
*                                                                     * 00900000
*                     STANDARD IMS USER EXIT PARMLIST                 * 00910000
*                   ------------                                      * 00920000
*              +0  ]            ]                                     * 00930000
*                  ]------------]                                     * 00940000
*              +4  ]            ]                                     * 00950000
*                  ]------------]      INPUT MESSAGE ROUTING EXIT     * 00960000
*              +8  ]            ]      PARMLIST                       * 00970000
*                  ]------------]        ------------------------     * 00980000
*              +12 ] A(PARMLIST)]-->+0  ] A(ECB)                 ]    * 00990000
*                   ------------        ]------------------------]    * 01000000
*                                   +4  ] A(SCD)                 ]    * 01010000
*                                       ]------------------------]    * 01020000
*                                   +8  ] FLAGS (SEE BELOW)      ]    * 01030000
*                                       ]------------------------]    * 01040000
*                                   +12 ] ZERO OR A(MESSAGE SEG) ]    * 01040600
*                                       ]------------------------]    * 01041000
*                                   +16 ] A(DESTINATION NAME)    ]    * 01041900
*                                       ]------------------------]    * 01042000
*                                   +20 ] A(DESTINATION LENGTH)  ]    * 01043000
*                                       ]------------------------]    * 01043900
*                                   +24 ] VARIABLE (SEE BELOW)   ]    * 01044000
*                                       ]------------------------]    * 01045000
*                                   +28 ] VARIABLE (SEE BELOW)   ]    * 01045900
*                                       ]------------------------]    * 01046000
*                                   +32 ] A(MSC MSNAME FIELD)    ]    * 01047000
*                                       ]------------------------]    * 01047900
*                                   +36 ] A(MSC SID FIELD)       ]    * 01048000
*                                        ------------------------     * 01049000
```

```
*                                                                     * 01050000
*              R13    = SAVE AREA ADDRESS                             * 01060000
*              R14    = RETURN ADDRESS                                * 01070000
*              R15    = ENTRY POINT ADDRESS                           * 01080000
*                                                                     * 01085000
*       CONTENTS OF INPUT MESSAGE ROUTING EXIT PARMLIST:              * 01090000
*                                                                     * 01100000
*           PARAMETERS FOR NORMAL INPUT MESSAGES, LU6.2 (APPC)        * 01110000
*           INPUT MESSAGES, AND OTMA INPUT MESSAGES:                  * 01120000
*                                                                     * 01130000
*              +0     = CURRENT ECB ADDRESS                           * 01140000
*                                                                     * 01142000
*              +4     = SCD ADDRESS                                   * 01145000
*                                                                     * 01147000
*              +8     = MESSAGE FLAG                                  * 01150000
*                                                                     * 01155000
*                     00000000 - MSG ORIG FROM NON-LU 6.2 AN   @PN64987 01160000
*                                NON-OTMA DEVICES/PROGRAM      @PN64987 01165000
*                     00000004 - MESSAGE ORIGINATED FROM LU 6.2       * 01170000
*                                PROGRAM/DEVICE.                      * 01172000
*                     00000008 - MSG ORIG FROM OTMA            @PN64987 01175000
*                                                                     * 01177000
*              +12    = ADDRESS OF THE FIRST MESSAGE SEGMENT IF       * 01180000
*                       MESSAGE ORIGINATED FROM NON-LU 6.2 OR         * 01180500
*                       NON-OTMA (IE: PARMLIST+8 = ZERO). ZERO        * 01181000
*                       IF LU6.2 OR OTMA MESSAGE                      * 01181600
*                                                                     * 01182000
*                       MESSAGE SEGMENT HAS THE FORMAT:               * 01182700
*                                                                     * 01183000
*                          LLZZDESTNAMEDATA                           * 01183800
*                                                                     * 01184000
*                          WHERE:                                     * 01184900
*                           LL = TWO BYTE SEGMENT LENGTH. THIS LENGTH * 01185000
*                                MUST NOT BE CHANGED                  * 01186000
*                           ZZ = SEGMENT FLAGS. DO NOT CHANGE.        * 01186600
*                           DESTNAME = DESTINATION NAME. DESTNAME     * 01187000
*                                      LENGTH IS PARAMETER AT         * 01187700
*                                      PARMLIST+20                    * 01188000
*                           DATA = REMAINING DATA IN THE SEGMENT      * 01188800
*                                                                     * 01189000
*              +16    = ADDRESS OF A 1 TO 8 BYTE DESTINATION NAME     * 01190000
*                       FIELD (DESTNAME), LEFT JUSTIFIED AND PADDED   * 01191000
*                       WITH BLANKS TO 8 BYTES. THE EXIT MAY   @PN67700 01192000
*                       CHANGE THE DESTINATION NAME. IF THIS   @PN67700 01193000
*                       NAME IS CHANGED, THE DESTNAME LENGTH   @PN67700 01195000
*                       POINTED TO BY THE PARMLIST+20 MAY NEED @PN67700 01196000
*                       TO BE CHANGED TO REFLECT THE LENGTH    @PN67700 01197000
*                       OF THE NEW DESTINATION NAME.           @PN67700 01198000
*                                                              @PN67700 01199000
*                                                                       01200000
*                                                                     * 01201000
*              +20    = ADDRESS OF A 4 BYTE FIELD WITH THE LENGTH OF  * 01202000
*                       THE DESTINATION NAME. IF THE DESTINATION      * 01203000
*                       NAME POINTED TO BY PARMLIST+16 IS CHANGED,    * 01205000
*                       THIS LENGTH SHOULD BE CHANGED TO REFLECT THE  * 01206000
*                       LENGTH OF THE NEW DESTINATION NAME.           * 01207000
*                                                                     * 01208000
*              +24    = ADDRESS OF A 1 TO 8 BYTE INPUT LUNAME, LEFT   * 01210000
*                       JUSTIFIED AND PADDED WITH BLANKS.             * 01212000
```

```
*                       IF THE INPUT MESSAGE IS FROM A NON-LU6.2 VTAM * 01214000
*                       PROGRAM/DEVICE (IE: BTAM) OR NON-OTMA CLIENT, * 01216000
*                       THE ADDRESS WILL POINT TO 8 BYTES OF ZEROS.   * 01218000
*                                                                     * 01220000
*                       FOR LU6.2 MESSAGES, ADDRESS OF A 1 TO 17 BYTE * 01230000
*                       NETWORK QUALIFIED INPUT LUNAME, LEFT          * 01235000
*                       JUSTIFIED AND PADDED WITH BLANKS TO 17 BYTES. * 01240000
*                                                                     * 01250000
*                       FOR OTMA MESSAGES, ADDRESS OF A 1 TO 8 BYTE   * 01260000
*                       TPIPE NAME LEFT JUSTIFIED AND PADDED WITH     * 01270000
*                       BLANKS TO 8 BYTES.                            * 01275000
*                                                                     * 01280000
*              +28    = ZERO FOR NON-LU6.2 AND NON-OTMA MESSAGES      * 01290000
*                                                                     * 01300000
*                       FOR LU6.2 MESSAGES, ADDRESS OF LU6.2 MESSAGE  * 01310000
*           OTMA ORIGINATED MESSAGE SPECIFIC PARAMETER:        @PN64987 01311500
*                                                                     * 01313000
*                       FOR OTMA MESSAGES, ADDRESS OF A 1 TO 16 BYTE  * 01314000
*              +20    = ADDRESS OF 8-BYTE TPIPE NAME           @PN64987 01314500
*              +24    = ADDRESS OF 16-BYTE MEMBER NAME         @PN64987 01316000
*                                                              @PN64987 01317500
*           OTHER TYPE MESSAGE SPECIFIC PARAMETERS:            @PN64987 01320000
*                       FIELD IS A TRANCODE. THE MESSAGE WILL BE      * 01330000
*                       REROUTED TO THE MSC LINK WHERE THE MSNAME IS  * 01340000
*                       ASSIGNED TO AND SENT TO THE DESIGNATED REMOTE * 01342000
*                       IMS SYSTEM. NO REROUTE IS DONE IF THE FIELD   * 01345000
*                       IS LEFT TO 0. THE 8 BYTE MSNAME MUST BE LEFT  * 01350000
*                       JUSTIFIED AND PADDED WITH BLANKS. THE MULTIPLE* 01355000
*                       SYSTEM COUPLING (MSC) FEATURE MUST BE         * 01360000
*                       AVAILABLE TO REROUTE THE MESSAGE.             * 01365000
*                                                                     * 01370000
*              +36    = ADDRESS OF 2 BYTES OF ZEROS THAT CAN BE       * 01380000
*                       CHANGED TO A MSC SYSTEM IDENTIFICATION (SID)  * 01390000
*                       THE MESSAGE WILL BE REROUTED TO THE MSC LINK  * 01395000
*                       WHERE THE SID IS ASSIGNED TO AND SENT TO THE  * 01398000
*                       DESIGNATED REMOTE IMS SYSTEM. THE MSC FEATURE * 01400000
*                       MUST BE AVAILABLE TO REROUTE THE MESSAGE.     * 01406000
*                       THE SID MUST BE A VALID VALUE (DEFINED AS     * 01410000
*                       REMOTE) BETWEEN 1-255 AND STORED IN THE LOW   * 01420000
*                       ORDER BYTE OF THE FIELD.                      * 01421000
*                                                                     * 01422000
*                       IF BOTH SID AND MSNAME ARE RETURNED, SID WILL * 01425000
*                       TAKE PRECEDENCE.                              * 01427000
*                                                                     * 01430000
*                       NOTE: IF AN INVALID MSNAME OR SID IS RETURNED,* 01440000
*                       THE REROUTE REQUEST WILL BE IGNORED AND THE   * 01441000
*                       TRANSACTION WILL BE ROUTED TO THE SYSTEM      * 01442000
*                       CURRENTLY ASSIGNED TO PROCESS THE TRANSACTION * 01443000
*                       (EITHER LOCAL OR REMOTE).                     * 01444000
*                                                                     * 01445000
*       CONTENTS OF REGISTERS AT EXIT:                                * 01446000
*                                                                     * 01447000
*              R15    = RETURN CODE ALWAYS ZERO                       * 01448000
*                                                                     * 01449000
*                                                                     * 01450000
*              DATA/OTHER : NONE                                      * 01460000
*                                                                     * 01470000
*   EXIT INTERFACES :                                                 * 01480000
*                                                                     * 01490000
```

```
*              REGISTERS AT EXIT (IF DIFFERENT FROM ENTRY) :          * 01500000
*                                                                     * 01510000
*                  R15  ZERO                                          * 01520000
*                                                                     * 01530000
*              RETURN CODES : R15 = 0                                 * 01540000
*                                                                     * 01550000
*                                                                     * 01560000
*   EXTERNAL ROUTINES CALLED : NONE                                   * 01570000
*                                                                     * 01580000
*   MESSAGE NUMBERS : NONE                                            * 01590000
*                                                                     * 01600000
*   ABEND CODES : NONE                                                * 01610000
*                                                                     * 01612000
*   CALLABLE SERVICES :                                               * 01615000
*                                                                     * 01617000
*       STORAGE SERVICES AND CONTROL BLOCK SERVICES CAN BE PERFORMED  * 01620000
*       BY INVOKING IMS CALLABLE SERVICES, REFER TO IMS               * 01630000
*       CUSTOMIZATION GUIDE: DATA COMMUNICATIONS FOR DETAILS.         * 01640000
*                                                                     * 01650000
*       THIS EXIT CAN USE CALLABLE SERVICES WITH THE ECB PASSED AT    * 01660000
*       +0 OF THE INPUT PARAMETER LIST                                * 01670000
*                                                                     * 01680000
*       THE FOLLOWING IS A DIRECTORY OF THE CONTROL BLOCKS  AND THEIR * 01690000
*       ASSOCIATED FIELDS THAT ARE INTENDED FOR ACCESS BY THE EXIT    * 01700000
*       VIA CALLABLE SERVICE:                                         * 01710000
*                                                                     * 01715000
*       CTB   CTBFLAG1 (CTB1CON, CTB1MAST, CTB1SUBP)                  * 01720000
*       CTT   CTTDEVIC                                                * 01730000
*       CLB   CLBNAME                                                 * 01740000
*                                                                     * 01750000
*       SMB   SMBDEQCT SMBENQCT SMBNAME SMBSTATS (SMBSRESP SMBSMULT   * 01760000
*             SMBSNOQU SMBNOSC SMBLOCK SMBSINQU SMBSQERR) SMBFLAG1    * 01770000
*             SMBFLAG2 (SMB2DRRT SMBFPPTX SMBFPXCL SMB2SMS SMB2RMT)   * 01780000
*             SMBFLAG3 (SMB3WFI) SMBPRIOR SMBCLASS SMBSPAL SMBLMTCT   * 01790000
*             SMBCOUNT SMBSIDR SMBSIDL SMBMXRGN SMBTCMDA SMBNOSCH     * 01800000
*             SMBPDIRN SMBRCTEN                                       * 01810000
*       PDIR  PDIRSYM PDIRCODE PDIROPTC PDIRBAD PDIRFLG3              * 01820000
*                                                                     * 01830000
*                                                                     * 01833000
*                                                                     * 01836000
*                                                                     * 01840000
*********************************************************************** 01850000
         EJECT                                                          01860000
DFSNPRTO CSECT                                                          01870000
DFSNPRTO AMODE 31                                                       01873000
DFSNPRTO RMODE ANY                                                      01876000
         SAVE  (14,12),,DFSNPRTO&SYSDATE&SYSTIME  SAVE REGISTERS        01880000
         LR    R12,R15                            ESTABLISH BASE        01890000
         USING DFSNPRTO,R12                                             01900000
         USING SCD,R11                                                  01905000
*                                                                       01910000
**********************************************************************  01920000
*                                                                    *  01930000
*  SET UP PARAMETERS FOR USER EXIT                                   *  01940000
*                                                                    *  01950000
**********************************************************************  01960000
*                                                                       01970000
         L     R15,12(,R1)         LOAD A(INPUT MSG ROUTING EXIT PARM)  01980000
         L     R10,0(,R15)         A(ECB)                               02020000
```

```
         L     R11,4(,R15)         A(SCD)                               02025000
         L     R1,12(,R15)         A(FIRST SEGMENT)                     02030000
         L     R2,16(,R15)         A(EIGHT BYTE DESTINATION NAME FIELD) 02040000
         L     R3,20(,R15)         A(4 BYTE LENGTH OF DEST NAME)        02050000
         L     R6,24(,R15)         A(INPUT LU NAME)                     02060000
         L     R8,32(,R15)         A(MSC MSNAME FIELD)                  02060000
         L     R9,36(,R15)         A(MSC SID FIELD)                     02060000
*                                                                       02070000
*                                                                       02080000
         L     R5,8(,R15)          MESSAGE FLAG                         02085000
         LTR   R5,R5               NON-LU 6.2 ORIGINATED MESSAGE?       02090000
         BZ    RMTRAN              NO, IT IS AN LU 6.2 ORIGINATED MSG   02095000
         L     R7,28(,R15)         A(DFSMSGRT)                          02100000
RMTRAN   DS    0H                                                       02110000
         CLI   SEVEN(R6),C'1'      IS THIS '1' LU                       02120000
         BNE   CHK2                                                     02130000
         MVC   ZERO(EIGHT,R8),MSN34N1                                   02300000
         MVC   ZERO(TWO,R9),SIDR41
         B     FINI
CHK2     DS    0H
         CLI   SEVEN(R6),C'2'      IS THIS '2' LU                       02120000
         BNE   CHK3                                                     02130000
         MVC   ZERO(EIGHT,R8),MSN34N2                                   02300000
         MVC   ZERO(TWO,R9),SIDR42
         B     FINI
CHK3     DS    0H
         CLI   SEVEN(R6),C'3'      IS THIS '3' LU                       02120000
         BNE   CHK4                                                     02130000
         MVC   ZERO(EIGHT,R8),MSN34N3                                   02300000
         MVC   ZERO(TWO,R9),SIDR43
         B     FINI
CHK4     DS    0H
         CLI   SEVEN(R6),C'4'      IS THIS '4' LU                       02120000
         BNE   CHK5                                                     02130000
         MVC   ZERO(EIGHT,R8),MSN34N4                                   02300000
         MVC   ZERO(TWO,R9),SIDR44
         B     FINI
CHK5     DS    0H
         CLI   SEVEN(R6),C'5'      IS THIS '5' LU                       02120000
         BNE   CHK6                                                     02130000
         MVC   ZERO(EIGHT,R8),MSN34N5                                   02300000
         MVC   ZERO(TWO,R9),SIDR45
         B     FINI
CHK6     DS    0H
         CLI   SEVEN(R6),C'6'      IS THIS '6' LU                       02120000
         BNE   CHK7                                                     02130000
         MVC   ZERO(EIGHT,R8),MSN34N6                                   02300000
         MVC   ZERO(TWO,R9),SIDR46
         B     FINI
CHK7     DS    0H
         CLI   SEVEN(R6),C'7'      IS THIS '7' LU                       02120000
         BNE   CHK8                                                     02130000
         MVC   ZERO(EIGHT,R8),MSN34N7                                   02300000
         MVC   ZERO(TWO,R9),SIDR47
         B     FINI
CHK8     DS    0H
         CLI   SEVEN(R6),C'8'      IS THIS '8' LU                       02120000
         BNE   CHK9                                                     02130000
         MVC   ZERO(EIGHT,R8),MSN34N8                                   02300000
```

```
         MVC   ZERO(TWO,R9),SIDR48
         B     FINI
CHK9     DS    0H
         CLI   SEVEN(R6),C′9′      IS THIS ′9′ LU                      02120000
         BNE   CHK0                                                    02130000
         MVC   ZERO(EIGHT,R8),MSN34N9                                  02300000
         MVC   ZERO(TWO,R9),SIDR49
         B     FINI
CHK0     DS    0H
         MVC   ZERO(EIGHT,R8),MSN34N0                                  02300000
         MVC   ZERO(TWO,R9),SIDR40
FINI     LM    R14,R12,12(R13)     RESTORE REGISTERS                   02320000
         SR    R15,R15             SET ZERO RETURN CODE                02330000
         BR    R14                 RETURN TO CALLER                    02340000
*                                                                      02350000
*                                  CONSTANTS AND SYMBOLS               02360000
*                                                                      02370000
CONST17  DC    H′17′               L(TRANCODE+REMOTE+LLXX)-1           02380000
DSFF     DC    CL4′DSFF    ′       DSFF- TRAN CODE
MSN34N0  DC    CL8′MSN34N0 ′       MSC MSNAME FlD FOR LNK34N0
MSN34N1  DC    CL8′MSN34N1 ′       MSC MSNAME FlD FOR LNK34N1
MSN34N2  DC    CL8′MSN34N2 ′       MSC MSNAME FlD FOR LNK34N2
MSN34N3  DC    CL8′MSN34N3 ′       MSC MSNAME FlD FOR LNK34N3
MSN34N4  DC    CL8′MSN34N4 ′       MSC MSNAME FlD FOR LNK34N4
MSN34N5  DC    CL8′MSN34N5 ′       MSC MSNAME FlD FOR LNK34N5
MSN34N6  DC    CL8′MSN34N6 ′       MSC MSNAME FlD FOR LNK34N6
MSN34N7  DC    CL8′MSN34N7 ′       MSC MSNAME FlD FOR LNK34N7
MSN34N8  DC    CL8′MSN34N8 ′       MSC MSNAME FlD FOR LNK34N8
MSN34N9  DC    CL8′MSN34N9 ′       MSC MSNAME FlD FOR LNK34N9
SIDR40   DC    H′40′               MSC REMOTE SYSID=40
SIDR41   DC    H′41′               MSC REMOTE SYSID=41
SIDR42   DC    H′42′               MSC REMOTE SYSID=42
SIDR43   DC    H′43′               MSC REMOTE SYSID=43
SIDR44   DC    H′44′               MSC REMOTE SYSID=44
SIDR45   DC    H′45′               MSC REMOTE SYSID=45
SIDR46   DC    H′46′               MSC REMOTE SYSID=46
SIDR47   DC    H′47′               MSC REMOTE SYSID=47
SIDR48   DC    H′48′               MSC REMOTE SYSID=48
SIDR49   DC    H′49′               MSC REMOTE SYSID=49
*                                                                      02410000
SEGMENT  DSECT                     STANDARD INPUT SEGMENT              02420000
LEN      DS    H                   SEGMENT LENGTH                      02430000
ZZ       DS    H
TEXT     DS    0H
ZERO     EQU   0
ONE      EQU   1
TWO      EQU   2
THREE    EQU   3
FOUR     EQU   4
FIVE     EQU   5
SIX      EQU   6
SEVEN    EQU   7
EIGHT    EQU   8
NINE     EQU   9
DSTRT    EQU   12
         EJECT
         REQUATE
         EJECT
         PRINT NOGEN
```

```
         ISCD  SCDBASE=0
         END
/*
//LNKS     EXEC PGM=IEWL,
//            REGION=2048K,
//            PARM=('SIZE=(2048K,64K)',NCAL,LET,
//            REUS,XREF,LIST)
//SYSPRINT DD SYSOUT=A,
//         DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),
//         SPACE=(605,(10,10),RLSE,,ROUND)
//SYSPUNCH DD DSN=IMSPERF.IMS510.MOBJDSET,DISP=SHR
//SYSLMOD  DD DSN=IMSPERF.IMS510.MEXITS3,DISP=SHR
//SYSUT1   DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSPUNCH)),
//         SPACE=(CYL,(10,1))
//SYSLIN   DD *
  INCLUDE SYSPUNCH(DFSNPRT0)
  NAME DFSNPRT0(R)
/*
```

Appendix C. Special Notices

This publication is intended to help system programmers, capacity planning specialists, and system administrators understand, plan, and implement an IMS multiple systems coupling environment as well as diagnose problems that may arise. The information in this publication is not intended as the specification of any programming interfaces that are provided by IMS/ESA and DB2. See the PUBLICATIONS section of the IBM Programming Announcement for IMS/ESA and DB2 for more information about which publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ACF/VTAM
AS/400
AT
BookManager
DB2
ES/9000
ESCON
IBM
IMS
IMS/ESA
MVS/ESA
NetView
Parallel Sysplex
RACF
RMF
RS/6000
S/390
System/390
VTAM

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Java and HotJava are trademarks of Sun Microsystems, Inc.

Other trademarks are trademarks of their respective companies.

Appendix D. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

D.1 International Technical Support Organization Publications

IMS/ESA Version 5.1 Guide, GG24-4302

D.2 Related Publications

These IBM publications are also relevent as further information sources:

DB2 for MVS/ESA V4 Administration Guide, SC26-3265-00

IMS/ESA V5 Operations Guide, SC26-8029-00

IMS/ESA V5 Operator's Reference, SC26-8030-00

IMS/ESA V5 Utilities Reference: System, SC26-8035-00

IMS/ESA Workload Router Licensed Program Specification, GC26-8946-00

IMS/ESA Workload Router User Guide, SC26-8945-01

IMS/ESA V5 Administration Guide: Transaction Manager, SC26-8013-00

IMS/ESA V5 Customization Guide, SC26-8020-00

IMS/ESA V5 Diagnosis Guide and Reference, LY27-9620-00 (available to IBM licenced customers only)

IMS/ESA V5 Installation Volume 2: System Definition and Tailoring, SC26-8024-01

D.3 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	**Subscription Number**	**Collection Kit Number**
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL `http://www.redbooks.ibm.com`.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type `GOPHER.WTSCPOK.ITSO.IBM.COM`
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BookManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

  `http://w3.itso.ibm.com/redbooks`

- **IBM Direct Publications Catalog on the World Wide Web**

  `http://www.elink.ibmlink.ibm.com/pbl/pbl`

  IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to `announce@webster.ibmlink.ibm.com` with the keyword `subscribe` in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications
Publications Customer Support
P.O. Box 29570
Raleigh, NC 27626-0570
USA

IBM Publications
144-4th Avenue, S.W.
Calgary, Alberta T2P 3N5
Canada

IBM Direct Services
Sortemosevej 21
DK-3450 Allerød
Denmark

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
(+45) 48 14 2207 (long distance charge)	Outside North America

- **1-800-IBM-4FAX (United States)** or **(+1)001-408-256-5422 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to `softwareshop@vnet.ibm.com`
- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to `announce@webster.ibmlink.ibm.com` with the keyword `subscribe` in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	**Order Number**	**Quantity**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number
- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

List of Abbreviations

APA	All Points Addressable
ACB	To VSAM, the access control block; to IMS, the application control block.
ACF/VTAM	Advanced Communication Function/Virtual Telecommunications Access Method
APPC	Advanced Program-To-Program Communication
APPLID	VTAM Application ID
BGWRT	Back Ground Write (IMS VSAM feature)
BLDS	Macro to build directories in memory of partitioned data sets
BMP	Batch Message Processing (IMS region)
BSC	Binary Synchronous
BTAM	Basic Telecommunications Access Method
CCB	Conversation Control Block
CF	Coupling Facility
CIMS	RACF IMS Command Class
CNT	Communication Name Block
COS	Class of Service
CQS	Common Queue Server
CRB	Communications Restart Block
CSA	Common Storage Area
CTB	Communications Terminal Block
CTC	Channel to Channel
CVB	Command Verb Block
DASD	Direct Access Storage Device
DBD	Database Definition
DBDGEN	Database Definition Generation
DBRC	IMS/ESA Database Recovery Control
DDM	Device Dependent Module
ddname	Data Definition Name
DEDB	Data Entry Database
DDF	Distributed Data Facility
DFW	DASD Fast Write
DDM	Device Dependent Module
DIMS	RACF IMS Command Group Class
DMB	Data Management Block (what the DBD becomes in the IMS Control region)
ECB	Event Control Block
ECSA	Extended CSA (common storage area)
EMH	Expedited Message Handler
ESA	Enterprise Systems Architecture
ESDS	Entry Sequenced Data Set
ESCON	Enterprise Systems Connection
ETO	Extended Terminal Option
FES	Front End Switch
FIFO	First-In-First-Out
FP	Fast Path
GIMS	RACF IMS Transaction Group Class
GN	Get Next
GTF	Generalized Trace Facility
GU	Get Unique
HPTS	Highly Parallel Transaction System
IBM	International Business Machines Corporation
IOPCB	Input/Output Program Communication Block
IMS	Information Management System
IPCS	Interactive Problem Control System
IRLM	IMS Resource Lock Manager
ISC	Intersystems Communication
ISRT	Insert DL/I Call
ITASK	IMS Task
ITR	Internal Throughput Rate
JCL	Job Control Language
LGMGS	Long Message Queue Data Set, component of message queues
LLB	Logical Link Block

LNB	Logical Link Name Block
LTERM	Logical Terminal
LU	Logical Unit (VTAM)
LXB	Link Extension Block
MFS	Message Format Service
MPP	Message Processing Program
MSC	Multiple Systems Coupling
MSDB	Main Storage Database
MTM	Main Storage to Main Storage
MTO	Master Terminal Operator
MVS	Multiple Virtual Storage
NCP	Network Control Program
N-WAY	Sharing of IMS databases in a sysplex
ODF	Offline Dump Formatter
OLDS	Online Log Data Set
OTMA	Open Transaction Manager Access
PARMLIB	MVS Parameter Library Data Set
PCB	Program Communication Block
PFK	Program Function Key
PI	Program Isolation
PIU	Path Information Units
PROCLIB	Procedure library data set IMS procedures are in IMSESA.PROCLIB, and MVS procedures are in SYS1.PROCLIB.
PSB	Program Specification Block
PST	Partition Specification Table
PTERM	Physical Terminal
QBLKS	Queue Blocks Data Set, component of message queues
QMGR	IMS Queue Manager
RACF	Resource Access Control Facility
RDR	Remote Destination Routing
RECANY	Receive Any
RMF	Resource Management Facility
RPL	Request Parameter List
RSMB	Remote Scheduler Message Block
RTT	Resource Translate Table
RU	Request Unit
SCD	System Contents Directory
SDEP	Sequential Dependents
SDLC	Synchronous Data Link Control
SHMGS	Short Message Queue Data Set, component of message queues
SLU	Secondary Logical Unit (VTAM)
SLUx	System Logical Unit, x = 1, 2, and P
SMB	Scheduler Message Block
SQG	Shared Queues Group
SMU	Security Maintenance Utility
SPA	Scratch Pad Area
STA	System Targeted Affinities
STSN	Set and Test Sequence Number
SVC	Supervisor Call Instruction on System/390
TCB	Task Control Block
TCT	Transaction Class Table
TIMS	RACF IMS transaction class
TMR	Transaction Router Prefix
TPNS	Teleprocessing Network Simulator
TPPCB	Input/Output Program Communication Block, also called IOPCB
VTAM	Virtual Telecommunications Access Method
VTCB	VTAM Terminal Control Block
UTC	Coordinated Universal Time
VSO	Virtual Storage Option
WADS	Write Ahead Data Set
XRF	IMS Extended Recovery Facility

Index

ITSO Redbook Evaluation

IMS/ESA Multiple Systems Coupling in a Parallel Sysplex
SG24-4750-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redeval@vnet.ibm.com

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction ____________

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:
____

____

____

____

What other redbooks would you like to see published?
____

____

____

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**
____

____

____

____

____

Printed in U.S.A.

SG24-4750-00